RADC-TR-77-102
Final Technical Report
March 1977

IMAGE UNDERSTANDING AND INFORMATION EXTRACTION

Purdue University

Sponsored by
Defense Advanced Research Projects Agency (DoD)
ARPA Order No. 2893

Approved for public release;
distribution unlimited.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U. S. Government.

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

This report contains a few illustrations which are not of the highest printing quality but because of economical consideration, it was determined in the best interest of the government that they be used in this publication.

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

This report has been reviewed and approved for publication.

APPROVED: David J. Brazil

DAVID J. BRAZIL, Capt, USAF
Project Engineer

Do not return this copy. Retain or destroy.

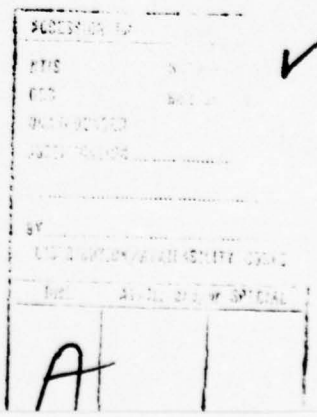
IMAGE UNDERSTANDING AND INFORMATION EXTRACTION

Thomas S. Huang
King-Sun Fu

Contractor: Purdue University
Contract Number: F30602-75-C-0150
Effective Date of Contract: 1 November 1975
Contract Expiration Date: 31 October 1976
Short Title of Work: Image Understanding and Information Extraction
Program Code Number: 7D30
Period of Work Covered: Nov 75 - Oct 76

Principal Investigators: T. S. Huang & K. S. Fu
Phone: 317 493-3361
Project Engineer: Capt D. J. Brazil
Phone: 315 330-3175

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by Capt David J. Brazil (IRRE), Griffiss AFB NY 13441 under Contract F30602-75-C-0150.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-77-102	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): IMAGE UNDERSTANDING AND INFORMATION EXTRACTION	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report. 1 Nov 75 - 31 Oct 76
	6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): Thomas S. Huang, King-Sun Fu	8. CONTRACT OR GRANT NUMBER(s): F30602-75-C-0150, ARPA Order-2893
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Purdue University, School of Electrical Engineering, West Lafayette IN 47907	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61101E, B8930001
11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Advanced Research Projects Agency, 1400 Wilson Blvd, Arlington VA 22209	12. REPORT DATE: March 1977
	13. NUMBER OF PAGES: 178
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Rome Air Development Center (IRRE), Griffiss AFB NY 13441	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer:
Capt David J. Brazil (IRRE)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital image processing, pattern recognition, syntactic pattern recognition, image segmentation, texture analysis, shape analysis, context, clustering, FLIR imagery, target detection.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report summarizes the results of the research program on Image Understanding and Information Extraction at Purdue University supported by the Defense Advanced Research Projects Agency, under Contract No. F30602-75-C-0150 for the twelve month period 1 November 1975 to 31 October 1976.

The objective of our research is to achieve a better understanding of image structure and to use this knowledge to develop techniques for image analysis and processing tasks, especially information extraction. Our emphasis is on

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

292 000 88

syntactic decomposition and recognition of imagery based on scene analysis. It is our expectation that this research will form a basis for the development of technology relevant to military applications of machine extraction of information from aircraft and satellite imagery.

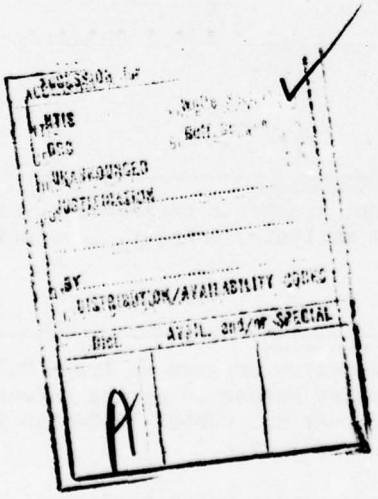


TABLE OF CONTENTS

IMAGE UNDERSTANDING AND INFORMATION EXTRACTION

RESEARCH SUMMARY

This is the final report of our research in Image Understanding and Information Extraction for the period 1 November 1975 to 31 October 1976. The objective of this research is to achieve better understanding of image structure and to improve the capability of image data processing systems to extract information from imagery and to convey that information in a useful form. The results of this research are expected to provide the basis for technology development relative to military applications of machine extraction of information from aircraft and satellite imagery.

Our research projects can be categorized into six rather heavily overlapping areas. Image Segmentation, Image Attributes, Image Structure, Image Recognition Techniques, Preprocessing and Applications. The relationships and interactions among these categories is suggested by Figure 1. After the sensor collects the image data, the preprocessor may either compress it for storage or transmission or it may attempt to put the data into a form more suitable for analysis. Image segmentation may simply involve locating objects in the image or, for complex scenes, determination of characteristically different regions may be required. Each of the objects or regions is categorized by the classifier which may use either classical decision-theoretic methods or some of the more recently developed syntactic methods. In linguistic terminology, the regions (objects) are primitives, and the classifier finds attributes for these primitives. Finally, the structural analyzer attempts to determine the spatial, spectral, and/or temporal relationships among the classified primitives. In some respects, this is where real "image understanding" is developed.

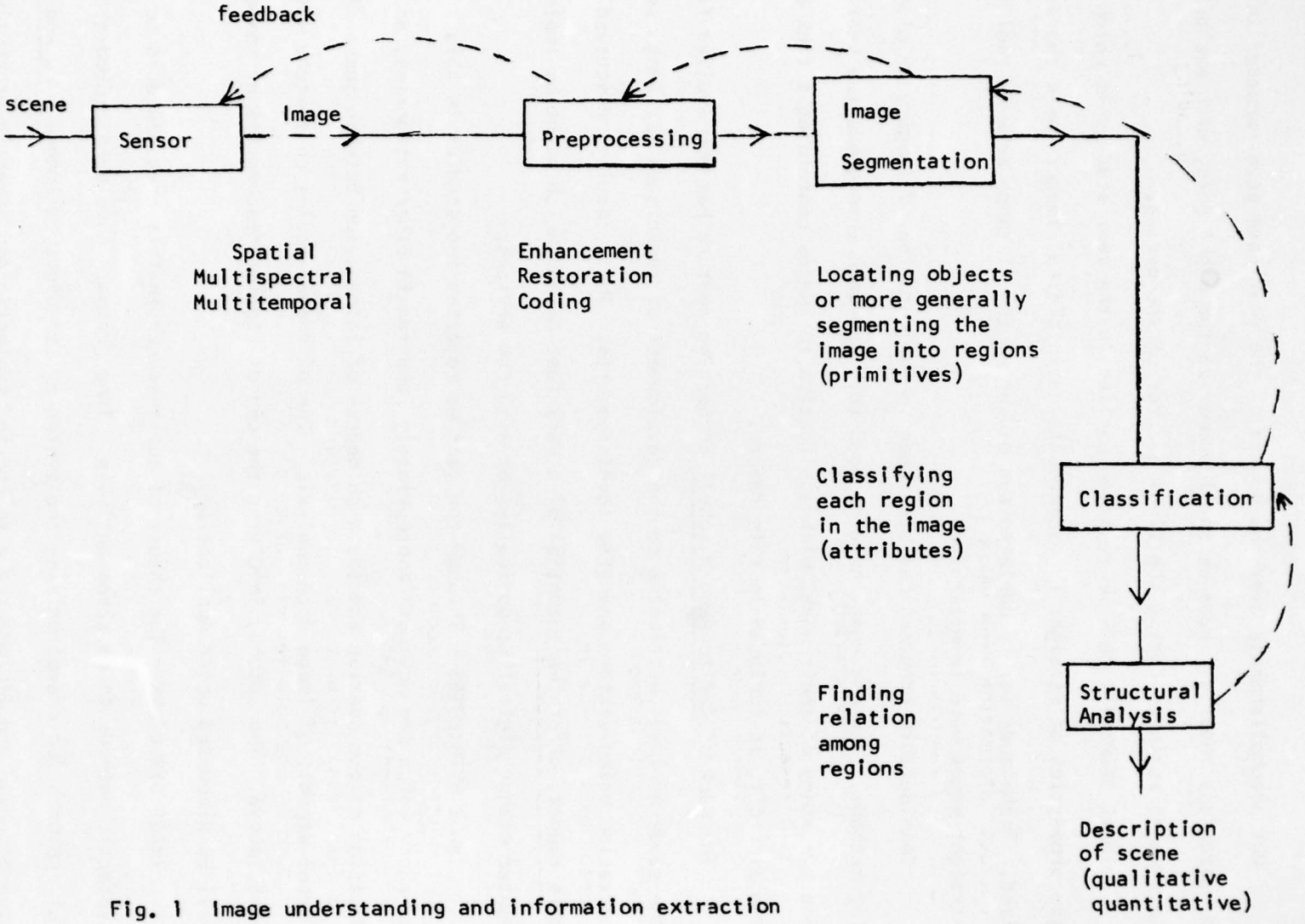


Fig. 1 Image understanding and information extraction

Our accomplishments over the course of the year have been recorded in our progress reports. Here we shall summarize the highlights, with emphasis on results achieved during the final quarter of the period.

IMAGE SEGMENTATION - We reported earlier in the year studies in which the properties of straight edges encountered in digital imagery were characterized. Techniques for recognizing and measuring the distance between real-life straight edges were investigated.

Considerable progress has been seen in segmentation of imagery by clustering methods. Yoo and Huang have pursued this approach throughout the year, and a summary of their work, with application to images containing a tank and two aircraft, is included in this report.

The work of Carlton and Mitchell concerning texture has now evolved from the study of image attributes to the development of techniques for image segmentation using texture and gray level features. Their results, discussed in this report, offer the potential of a very fast method which might be implemented either digitally, optically, or with CCD devices.

IMAGE ATTRIBUTES - Through the year we reported two studies in this category which are now more appropriately reported in other categories, an indication of the overlap and the high degree of interaction between these different aspects of image data analysis. One of these studies on texture was noted above. The second, involving the use of Fourier descriptors of shape, will be discussed under Applications.

IMAGE STRUCTURE - The thrust of our research in this area has been to use syntactic methods to do scene analysis. Tree grammars have been proved a useful approach for characterizing the syntax or structure of images. Fu and Keng have proposed and implemented a scheme for syntactic and semantic processing

of pre-classified data. In this report they show several examples of the application of their method for information extraction.

IMAGE RECOGNITION TECHNIQUES - A "supervised clustering" method has been shown by Fukunaga and Short to be useful for localizing a problem rather than dealing with a more difficult global problem. Computationally simple yet accurate results are obtained. Potential applications of the approach include linear classifier design and density estimation.

Fu and Yu and Swain and Pfaff have investigated somewhat different approaches to the utilization of context for classification. However, this research has not yet progressed far enough to support definite conclusions concerning the practical utility of the approach.

PREPROCESSING - Enhancement of blurred images by complex cepstral analysis is the goal of O'Connor and Huang. They report here further work on improving the reconstruction of the analytic phase function of the image. They are also interested in using this form of analysis for characterizing texture.

The "projection method" is an approach to image restoration which has been investigated by Berger and Huang. In this report they discuss the merits of this method compared to "singular value decomposition" and consider some aspects needing further development.

APPLICATIONS - Fourier descriptors have been demonstrated to be a useful means of describing the shape of a closed planar figure, and, in particular, Wallace and Wintz have used Fourier descriptors to encode the shapes of aircraft. They report here very good results using this approach for aircraft recognition.

Spatial filtering has been used by Mitchell, et. al. to reduce the

effects of light cloud cover in satellite imagery. Results from a computer simulation and from LANDSAT data are discussed in this report. Earlier in the year, preliminary results were reported of applying a combination of spatial frequency filtering and syntactic analysis for locating airports.

IMAGE SEGMENTATION BY CLUSTERING

M. Y. Yoo and T. S. Huang

I. Introduction

Photographic images include many different kinds of information, some of which may be relevant to a viewer's interests and others may attract some other viewers depending upon objectives and interests. But whatever the viewer's objectives are, the most interesting parts of images are high information carrying contours [1-3] which usually lie between parts of images with different characteristics to human visual perception. Therefore, the segmentation of images into more fundamental textural components is the first step in many image processing systems.

There could be two possible approaches for this purpose; one is the "top-to-bottom" approach. In this way we roughly segment the image into major textures and subdivide the textures into finer pictorial components until the results meet the desired objective. The other way is the "bottom-up" approach where the segmented major textures are results grown from the "seed textures". Each picture element is either merged into the texture already growing from a "seed" or provides a new "seed". The conventional segmentations are either one [4] or the other [2].

The new approach which we propose in this report is more flexible because the final segment is determined by the clustering in the feature plane rather than the original image. This approach could be either "bottom-up" or "top-to-bottom" depending upon whether we first assume many clusters and group them together or initially allow major clusters and take them apart later.

II. The Image Segmentation Algorithm

The image segmentation algorithm which we propose consists of three major steps: (1) feature pair extraction, (2) clustering in the feature pair joint

density plane, and (3) segmentation. The algorithm is described by the flow chart in Fig. 1.

The feature pair extraction is the measurement of parameters giving maximal information about the local properties of the images. Feature extraction, however, has proved too difficult to allow a general approach. Nevertheless, the extraction of significant features in a reasonably simple form is the most important step among the three.

The clustering of features in the feature pair joint density plane depends highly on the features extracted, and the better the features chosen, the easier the clustering.

The final step, segmentation, is the back transformation from clusters in the feature plane to the segmented image.

Some detailed descriptions about the three major steps of the algorithm were included in the previous report [5] and we will discuss new aspects here.

A. The Feature Pair Extraction

Interesting feature pairs should be chosen so as to give maximal information about the local textural properties in a reasonably simple form to implement on a digital computer.

Statistical and structural properties are the most commonly used. The local sample mean, local sample standard deviation, and the local minimum and maximum pairs were used for this report.

The feature pairs are calculated based on nonoverlapping 2×2 or overlapping 3×3 local windows. In the case of 2×2 nonoverlapping window the feature pairs are calculated based on the four grey levels at (x,y), $(x,y+1)$, $(x+1,y)$, and $(x+1,y+1)$ and the four picture points are represented by the calculated feature pairs. The same operations are performed in the next nonoverlapping window and we scan this window horizontally and vertically

until the whole image plane is covered. For 3 x 3 window case the pixel (x,y) is represented by the feature pairs calculated based on the grey level at (x,y) and the grey levels at the eight surrounding pixels.

The calculated feature pairs can be any non-negative values and to introduce the feature plane based on the extracted feature pair the features need to be normalized and quantized in appropriate levels. The features are normalized such that the maximum value of features is 64 and the feature is uniformly quantized into 64 levels. By counting pixels which have a particular feature pair we can calculate the joint density of each feature pair.

Figures 2 and 5 are feature planes based on the sample mean and sample standard deviation pair for the infrared (FLIR) image of a tank (Fig. 7(a) and an aerial photograph of an airplane (Fig. 7(b)). Feature planes based on the mean-standard deviation pair and the local min - local max pair for a second airplane (Fig. 7(c)) are shown in Fig. 6.

Computer line print outs such as shown in Fig. 3 include numerical information about the feature planes.

B. Clustering in the Feature Pair Joint Density Plane

There are several different clustering algorithms [6-10] but the size of the data set which we handle is too big to be implemented in an efficient way. Especially since we are clustering features rather than the original picture data and "nice" clusters in the feature plane do not lead to the "nice" segments in the image plane. According to our experience the most useful clustering has been "eyeball clustering" based on the numerical data in the feature plane (computer print out) in conjunction with the corresponding Gould Print.

Some textural components are separated from others mainly based on one feature and others are based on the other feature and some distance measures

in the feature plane, which many clustering algorithms are based on, do not mean much. In most cases our experiments the Gould print outs provide rough information about the number of clusters. The levels shown in Fig. 4 are the result of clusterings corresponding to the feature plane in Fig. 3. All "zero" levels in the cluster level plane denote feature points which the original image does not have.

C. Segmentation

This is the back-transformation of clusters in the feature plane onto the textural components in the original picture domain. Different clusters in the feature plane correspond to different textures in the original image domain and the number of textural components depends upon how many clusters we allow in the feature plane. In 2 x 2 winow case, if the feature pair based on the four grey levels at (x,y), ..., (x+1,y+1) is represented by level m in the feature plane, the same level m is assigned to the four pixels (x,y), ..., (x+1,y+1) of the original image, whereas the level n is given to the *only pixel* (x,y) *when the feature pair* based on the grey level at (x,y) and the grey levels at its eight neighbors has the level n for 3 x 3 window case.

But the clusters in the feature plane usually are not perfect (actually even if we achieve the ideal clustering we cannot avoid generating noise when some pixels in different textural components may happen to have the same feature pair in the feature plane) and a lot of noise is produced when the clusters are back-transformed onto the image domain. Therefore, some appropriate noise reduction processes without destroying the segmented major textures are highly desirable. The following two noise reduction processes are applied sequentially and the results turn out to be satisfactory.

1. Purge

As we see in part (a) of Figs. 8-13 the original segments (before noise reduction processes are applied) which are back-transformed from cluster levels are quite noisy. Some are due to imperfect clustering and some are due to the fact that some of different textural parts may have the same feature values. But most of the noise points are either small isolated clusters or horizontally or vertically oriented narrow strips with fairly short run length. The isolated noise will be discussed in the next section. Whether they are noise or not, it is quite reasonable that small narrow textural strips embedded in a large different textural component are merged into the dominant texture from the texture segmentation point of view. The following operation will clean up the horizontally oriented noisy strips.

Let $L(x,y)$ be the cluster level corresponding to the pixel (x,y) and $C(s) \equiv \{1,2,3,\ldots,S\}$,

$$\text{if } L(x-m,y) = L(x-n,y)$$

$$\text{and } L(x,y) \neq L(x-m,y) \text{ for some } m,n \in C(s)$$

$$\text{then } L(x,y) = L(x-m,y) .$$

For vertically oriented strips

$$\text{if } L(x,y-m) = L(x,y+n)$$

$$\text{and } L(x,y) \neq L(x,y-m) \text{ for some } m,n \in C(s)$$

$$\text{then } L(x,y) = L(x,y-m) .$$

The standard sizes of S are 1, 2, 3. If there are fairly narrow parts in the textural segments, it is recommended to choose smaller S,

otherwise some parts of the textures may be destroyed. For Figs. 8(b) and 9(b) S = 3 was used; S = 2 was used for Figs. 10(b) and 11(b); and S = 1 for Figs. 12(b) and 13(b). Even S = 2 is too big for Fig. 11(b). Note that the engine in the left side is connected to the body of airplane 2.

2. ISOLDROP

Most of the isolated noisy clusters are cleaned up by the "PURGE" but "PURGE" may also introduce some isolated noises. Of course, there are some fairly large clusters of noise which "PURGE" failed to clean.

We scan the m x n window and if all the grey levels at the boundary of the window are the same then we replace the whole window by the grey level at the boundary of the window. We used the size m = n = 5. If the small size of the window is chosen we cannot clean up fairly big size of noises but if we choose a big window size, noise clusters which are fairly close to each other may not be dropped. The compromise depends upon the type image involved.

III. Applications

A useful application for segmented textures is the boundary detection of the interesting objects in a closed form. Closed contours are important in relation to coding or shape description (pattern recognition). We pick up a pixel (x,y) in a particular texture (object) and look at its eight surrounding pixels. If at least one of the eight pixels belongs to any other texture we assume the pixel (x,y) is a boundary point. But before the boundary detection operation is applied there is one more thing to be considered.

If some isolated noisy clusters were located at boundaries between different textural components, the noise reduction operations "PURGE" and "ISOLDROP" could not handle the case. The following operation precedes

the boundary detection process to achieve clean boundaries. Let P be the level of the cluster corresponding to the interesting object in the image involved for horizontally oriented noise

if $L(x,y) = P$

and $L(x-m,y) \neq L(x,y)$

$L(x+n,y) \neq L(x,y)$ for some $m,n \in C(s)$

then $L(x,y) = L(x-m,y)$.

For vertically oriented noise

if $L(x,y) = P$

and $L(x,y-m) \neq L(x,y)$

$L(x,y+n) \neq L(x,y)$ for some $m,n \in C(s)$

then $L(x,y) = L(x,y-m)$.

Detected boundaries for three different images are shown in Figs. 8-13 (part (c)) and boundaries superimposed on the corresponding original images are shown in part (d) of the figures.

IV. Experimental Results and Discussion

We applied the algorithm to three different images: infrared image (tank) which has low resolution and two airplane images with moderate resolution. The size of the images are 256 x 256. Segmented images are displayed by a COMTAL 8000 Series System. Boundary detection based on the segmented textures guarantees the closed contours which are essential in many applications. All detected boundaries are based on the noise reduced segments except the airplane #2 (3 x 3 window case). In this case we used the

original noisy segments because the noise reduced version messed up the small parts near the left engine (see Fig. 11(b)). The boundaries superimposed on the originals demonstrate how accurate the algorithm segmented the original images.

REFERENCES

1. H. Freeman, "On the Encoding of Arbitrary Geometric Configurations," IRE Trans. on Electronic Computer, pp. 260-268, June 1961.

2. J. N. Gupta and P. A. Wintz, "A Boundary Finding Algorithm and Its Applications," IEEE Trans. on Circuit and System, Vol. CAS-22, No. 4, April 1975.

3. G. H. Granlund, "Fourier Preprocessing for Hand Print Character Recognition," IEEE Trans. on Computers, Vol. C-21, pp. 195-201, Feb. 1972.

4. R. L. Mattson and J. E. Dammann, "A Technique for Determining and Coding Subclasses in Pattern Recognition Problems," IBM Journal, pp. 294-302, 1965.

5. M. Y. Yoo and T. S. Huang, "Image Segmentation by Clustering," ARPA Interim Report F30602-75-C-0150, pp. 30-60, May-July 1976.

6. H. P. Friedman and J. Rubin, "On Some Invariant Criteria for Grouping Data," JASA, pp. 1159-1178, December 1967.

7. G. H. Ball and D. J. Hall, "A Clustering Technique for Summarixing Multivariate Data," Behavioral Science, Vol. 12, pp. 153-155, March 1967.

8. K. Fukunaga and W. Koontz, "A Criterion and an Algorithm for Grouping Data," IEEE Trans. on Computers, Vol. C-19, No. 10, pp. 917-923, Oct. 1970.

9. R. A. Jarvis and E. A. Patrick, "Clustering Using a Similarity Measure Based on Shared Near Neighbors," IEEE Trans. on Computers, Vol. C-22, No. 11, pp. 1025-1034, November 1973.

10. W. Koontz, P. M. Narendra, and K. Fukunaga, "A Graph Theoretic Approach to Non-parametric Cluster Analysis," IEEE Trans. on Computers, Vol. C-25, No. 9, pp. 936-944, September 1976.

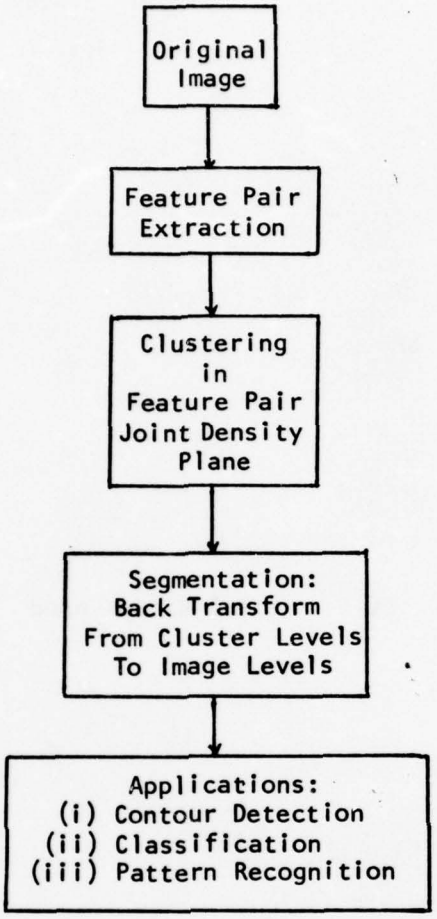
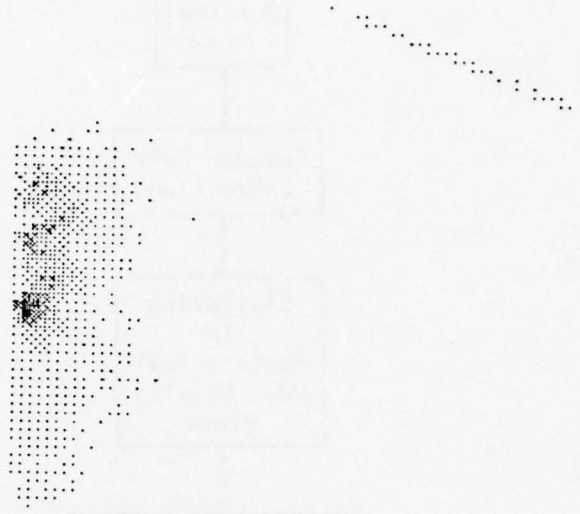


Figure 1 The Segmentation Algorithm



(A) 2 X 2 Window used

(B) 3 X 3 Window used

Figure 2 Feature Plane(Tank)
Ver: Mean Hor: Standard deviation
Size: 64 X 64

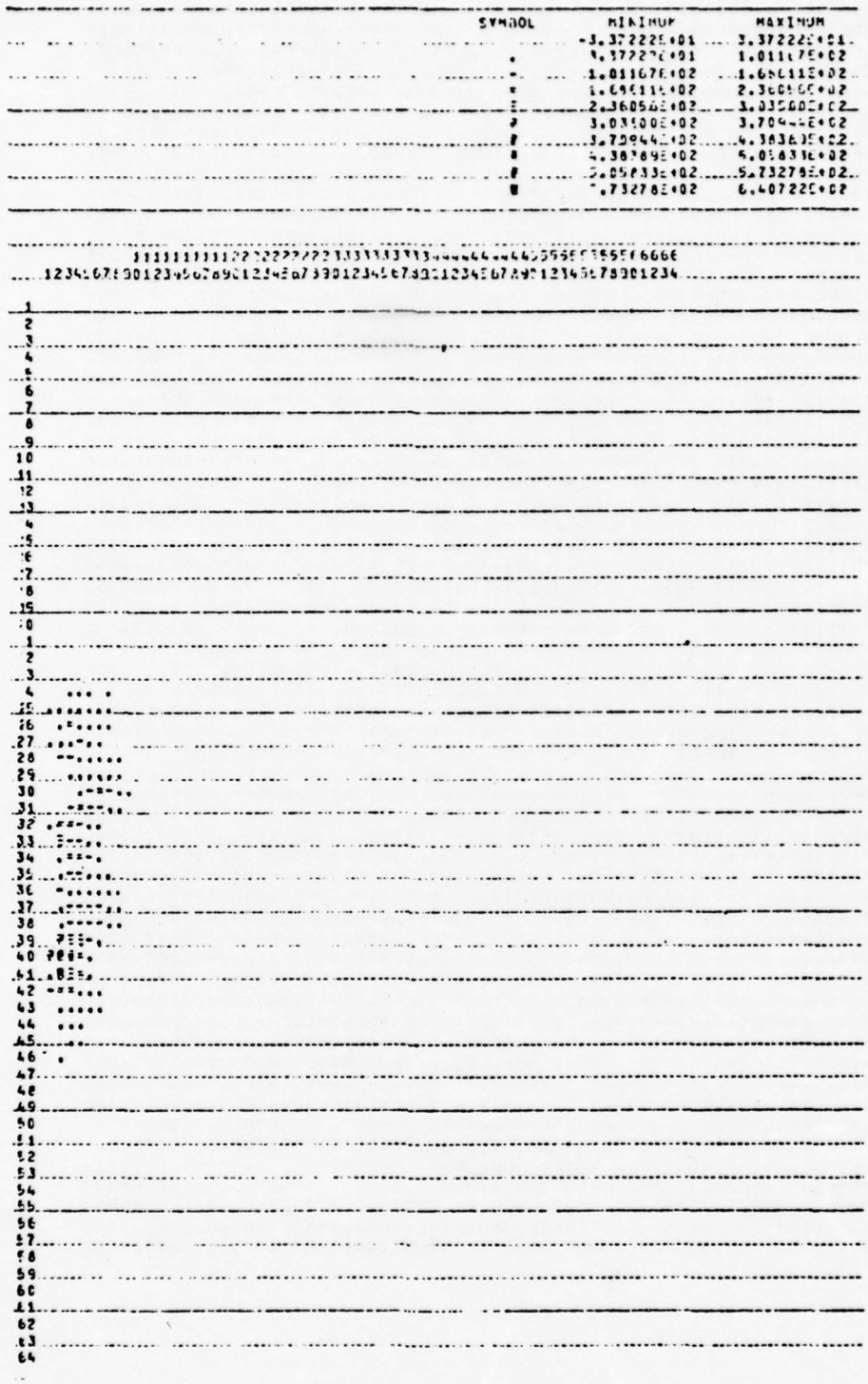


Figure 3 Feature Plane (Tank)
Ver: Mean Hor: Standard deviation
2 X 2 Window used.

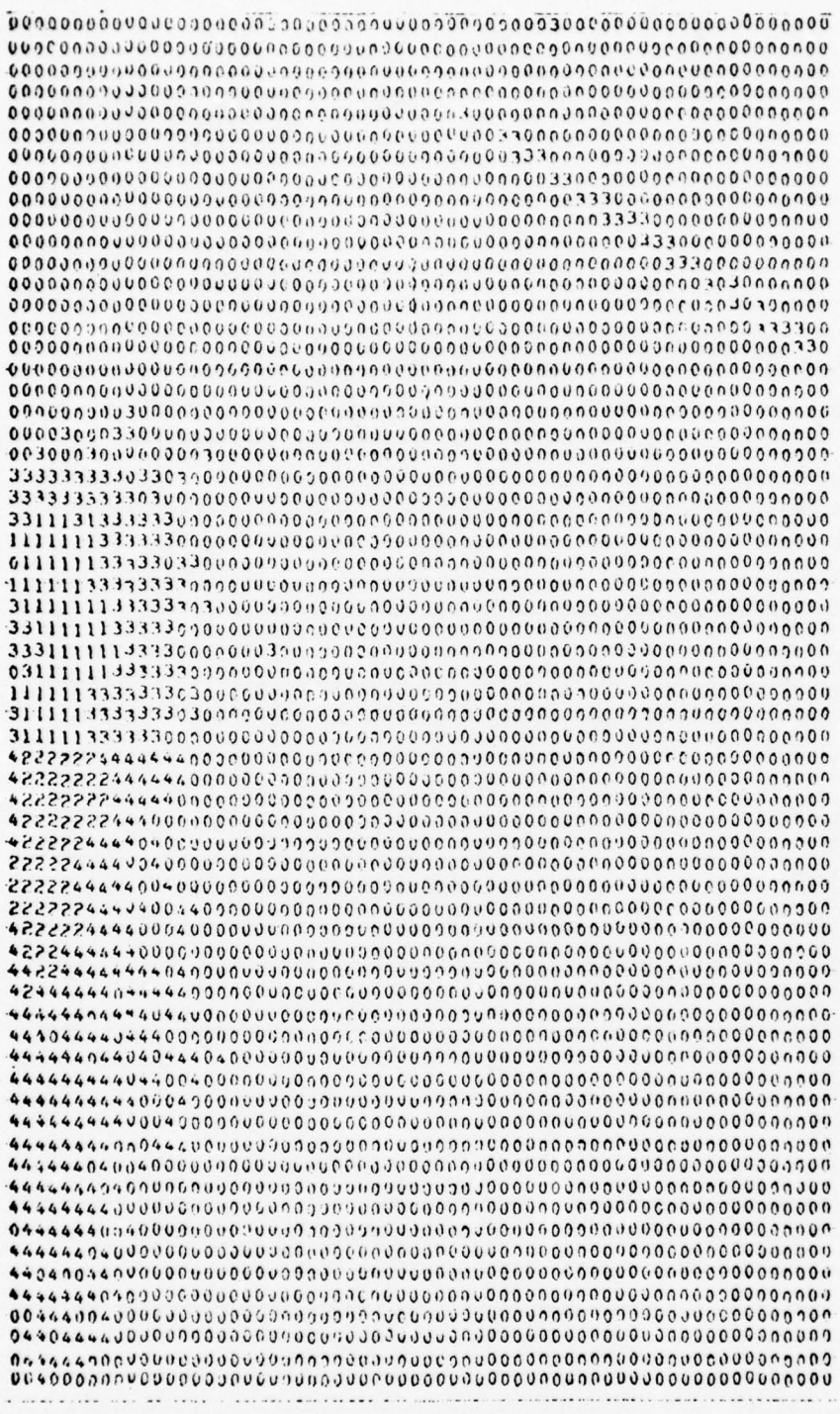
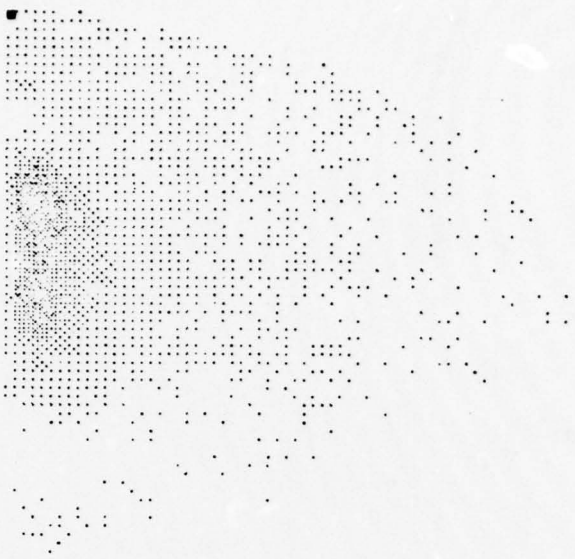
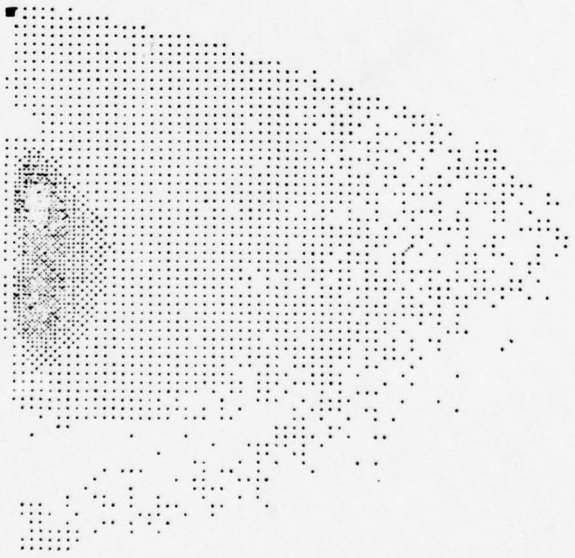


Figure 4 Levels of Clusters(Tank)
Feature(Ver: Mean Hor: Stand. devi.)
2 X 2 Window used.

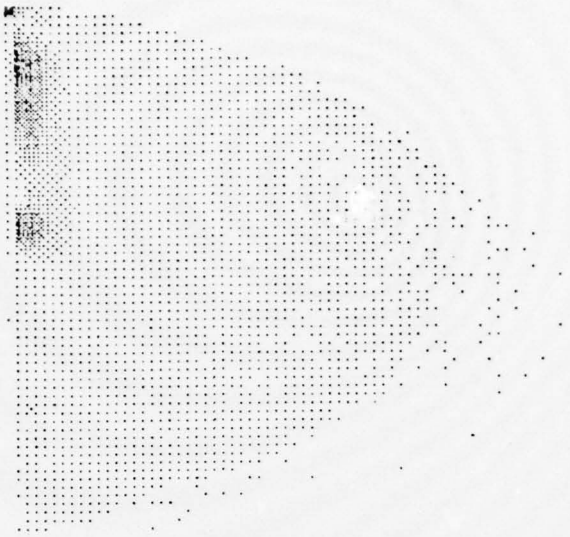


(A) 2 X 2 Window used

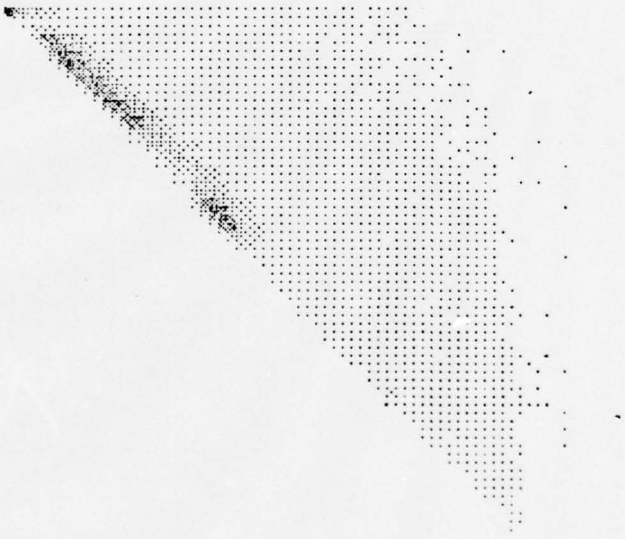


(B) 3 X 3 Window used

Figure 5 Feature Plane(Airplane 1)
Ver: Mean Hor: Standard deviation
Size: 64 X 64

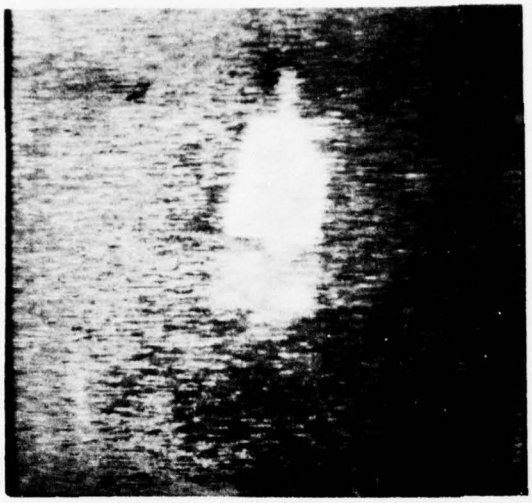


(A) Ver: Mean Hor: Standard deviation

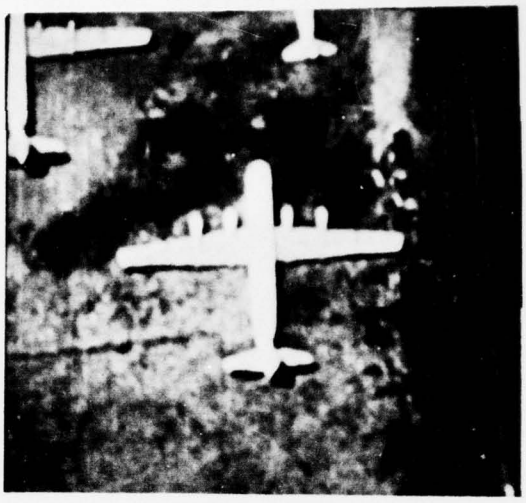


(B) Ver: Local minimum Hor: Local maximum

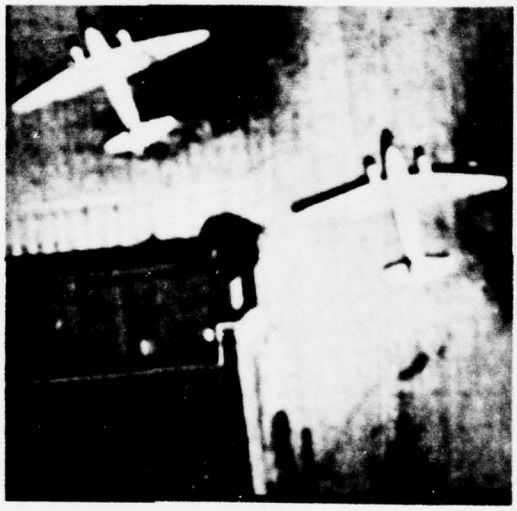
Figure 6 Feature Plane(Airplane 2)
Size: 64X64
3 X 3 Window used



(a)

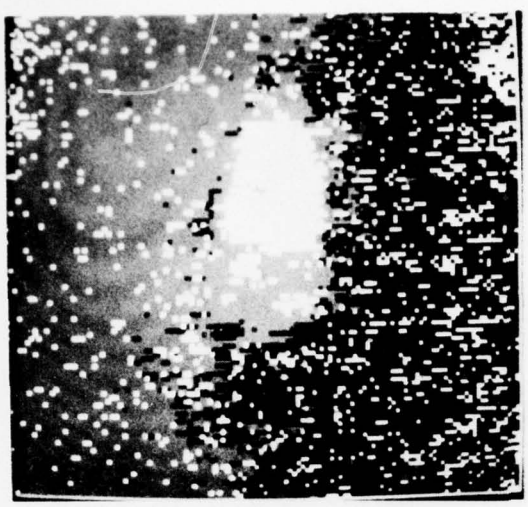


(b)

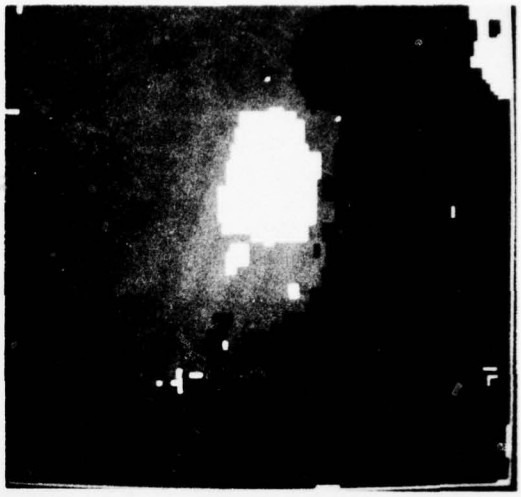


(c)

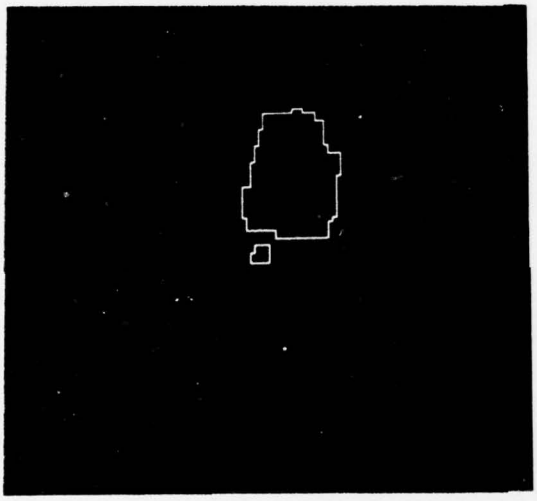
Figure 7 Original Pictures.
(a) FLIR tank
(b) Airplane 1
(c) Airplane 2



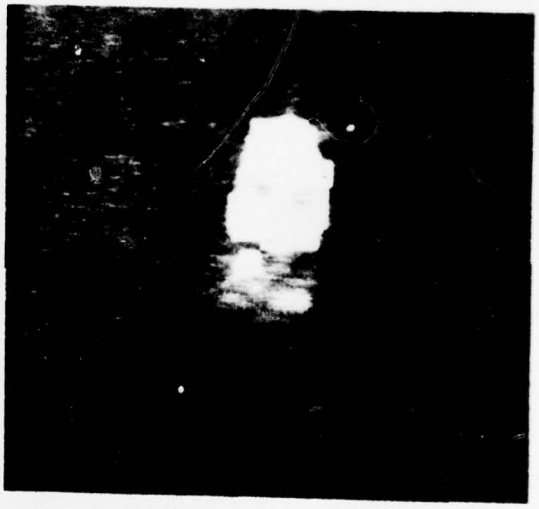
(a)



(b)



(c)



(d)

Figure 8 Results using Original in Fig. 7(a), a 2 x 2 Window, and the Mean and Standard Deviation Features.
(a) Segmented image
(b) Noise reduced segments
(c) Boundary of interesting object
(d) Boundary superimposed on original

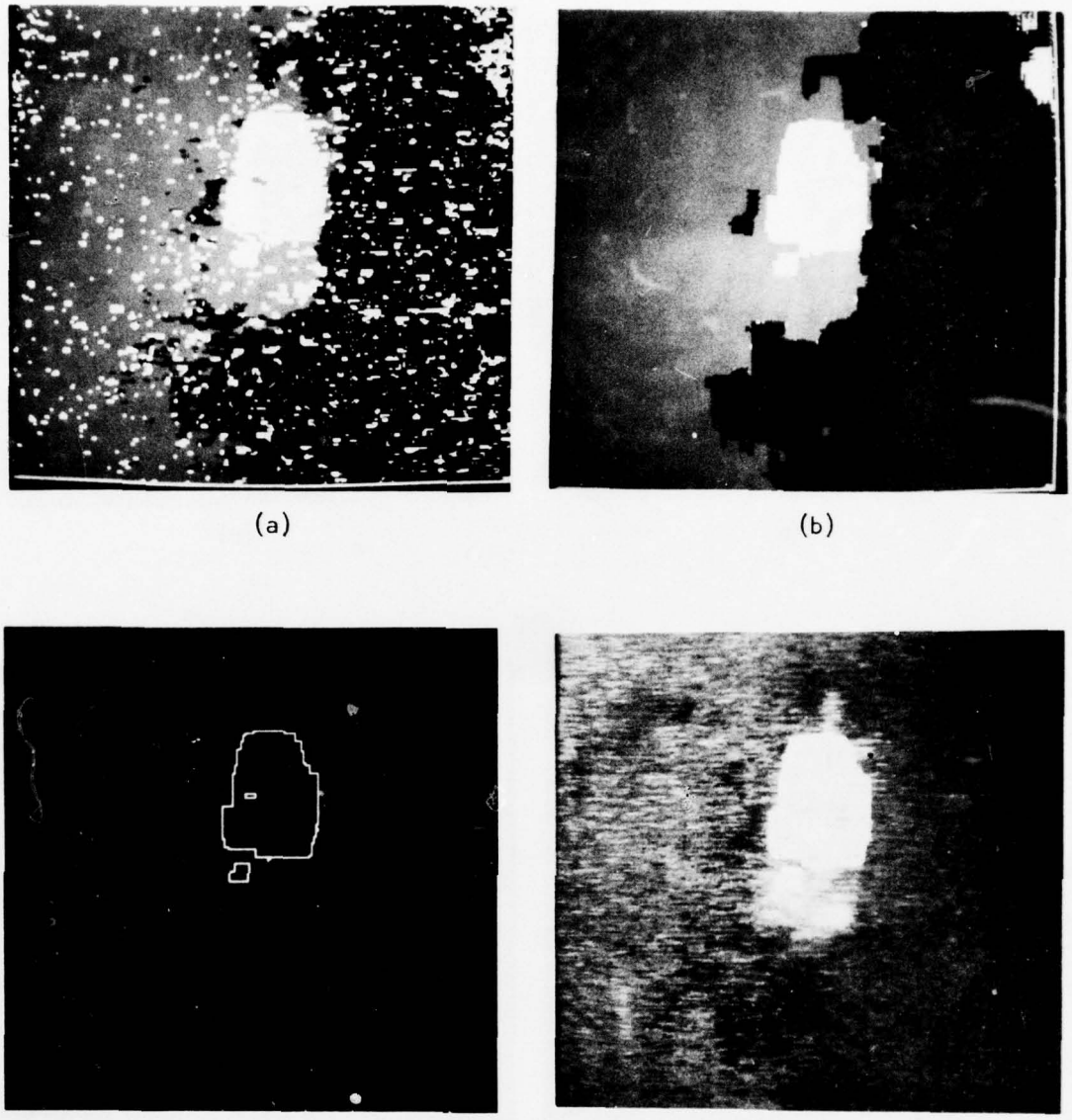


Figure 9 Same as Fig. 8 except a 3 x 3 Window.
(a) Segmented image
(b) Noise reduced segments
(c) Boundary of interesting object
(d) Boundary superimposed on original

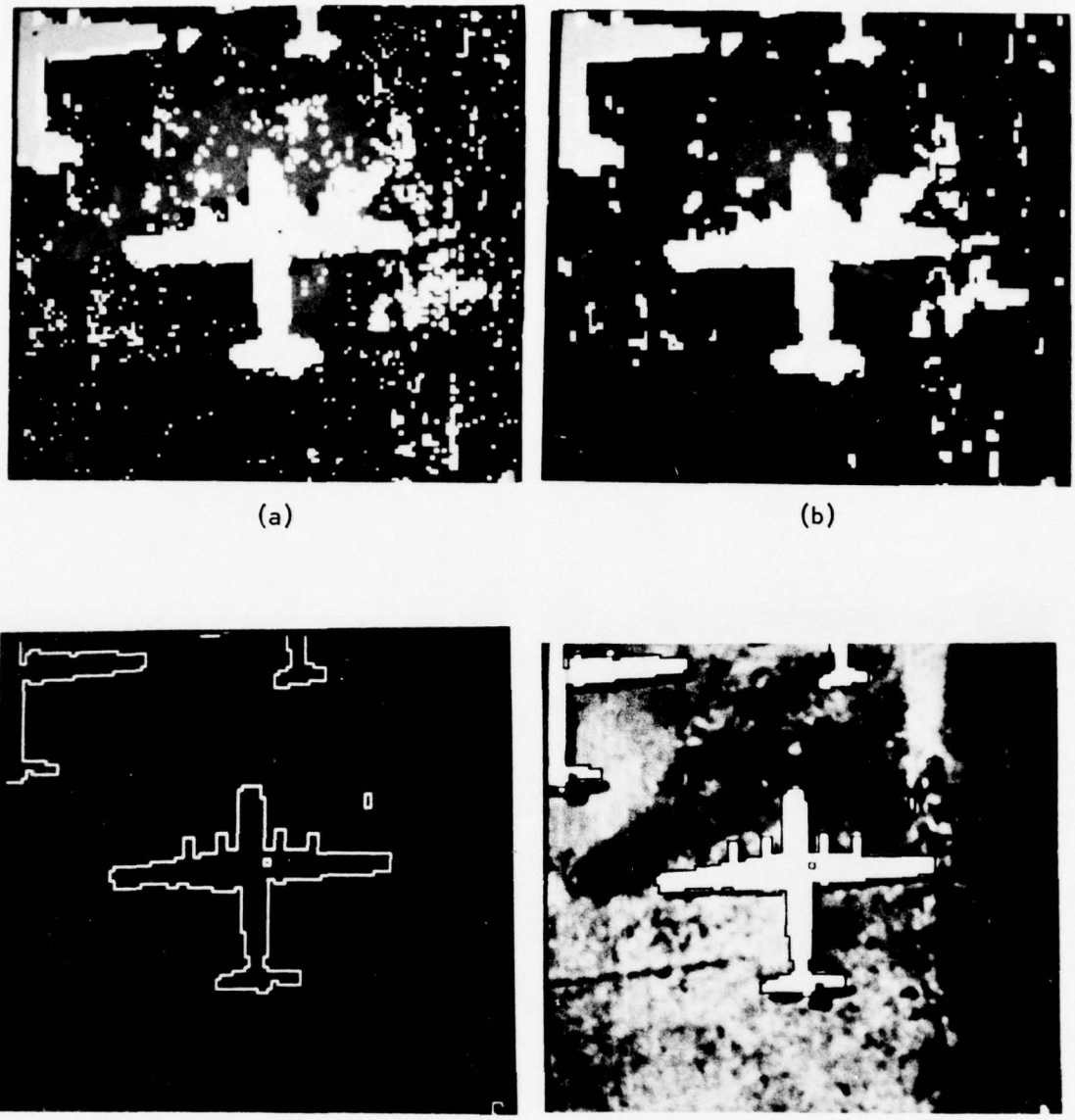


Figure 10 Results using Original in Fig. 7(b), a 2 x 2 Window, and the Mean and Standard Deviation Features.
(a) Segmented image
(b) Noise reduced segments
(c) Boundary of interesting objects
(d) Boundary superimposed on original

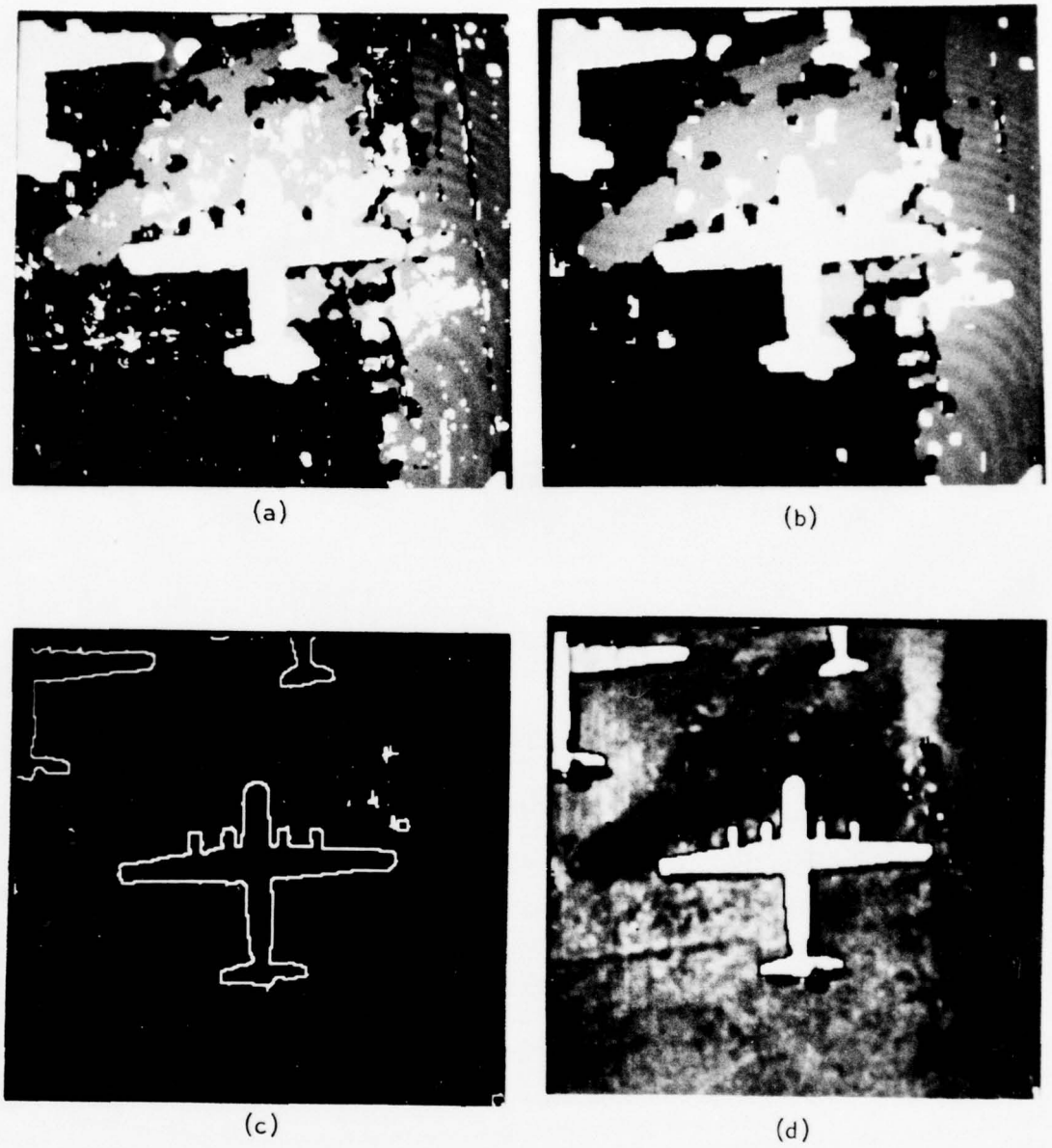


Figure 11 Same as Fig. 10 except a 3 x 3 Window was used.
(a) Segmented image
(b) Noise reduced segments
(c) Boundary of interesting objects
(d) Boundary superimposed on original

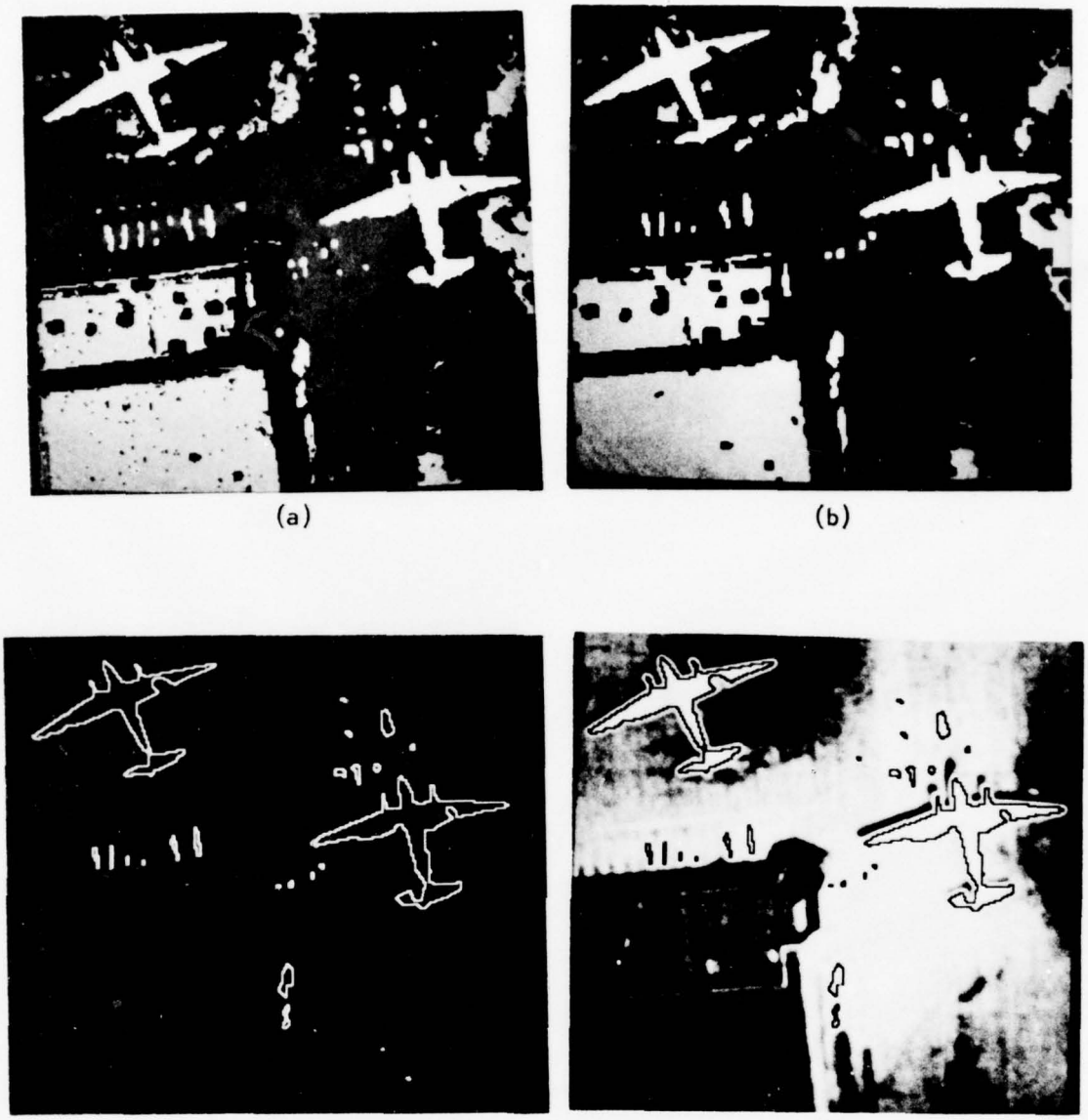
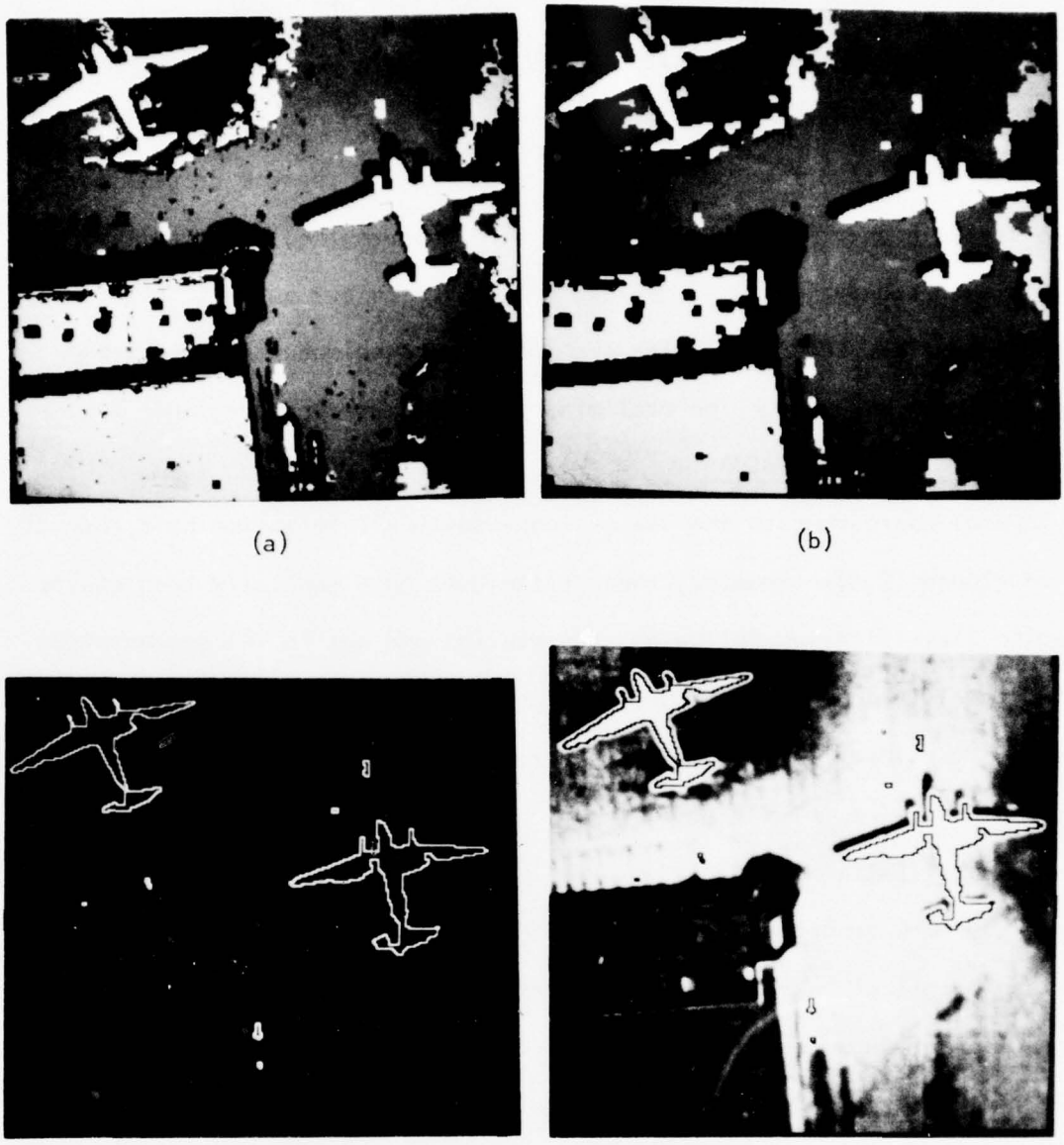


Figure 12 Results using the Original in Fig. 7(c), a 3 x 3 Window, and the Mean and Standard Deviation Features.
(a) Segmented image
(b) Noise reduced segments
(c) Boundary of interesting objects
(d) Boundary superimposed on original



(a) (b)

(c) (d)

Figure 13 Same as Fig. 12 except the Features used are Local Min and Local Max.
(a) Segmented image
(b) Noise reduced segments
(c) Boundary of interesting objects
(d) Boundary superimposed on original

IMAGE SEGMENTATION USING TEXTURE AND GRAY LEVEL

S.G. Carlton and O.R. Mitchell

I. INTRODUCTION

In our last report we described the use of image texture properties to locate region boundaries. While results were encouraging, the inability to obtain closed boundaries was a decided drawback. In our pursuit to produce closed boundaries, thus segmenting the image, we developed the algorithm described in this report. The preliminary results are encouraging.

II. THE BASIC TEXTURE MEASURE

Several approaches to the use of image texture information have recently been developed [2,3]. However, these techniques have generally been applied to region classification following segmentation and not to the segmentation problem itself. In our approach to represent texture, various sizes of local extrema in the logarithm of the image are summed within a window surrounding each point.

A. Local Extrema

In the work reported here, local extrema were found by combining horizontal and vertical one-dimensional operations. The one-dimensional operation scans a line of data and assigns a point to be a local maximum (minimum) of size T if it is the largest (smallest) value occurring in the vicinity on the line before the values drop (rise) to an amount T below (above) this maximum (minimum) value [4]. An example is shown in Figure 1. The local extrema of size 3 and size 1 are marked. This process is equivalent to detecting the extreme following a hysteresis smoothing operation using a smoothing of T/2.

The logarithm operation is first performed on the image prior to the extrema detection. The use of this texture measure is a crude attempt to simulate the human visual system's response to a texture pattern. For example,

each maximum in Figure 1 would appear as a bright point to a human observer even though one of them is below a minimum which would appear dark to an observer, i.e., the local surround affects perceived brightness much more than the actual gray level [5].

A sample image is shown in Figure 2. This is a 256x256 8-bit black and white aerial scene of a military simulation area in New York State. The local extrema measured in the logarithm version of this image are shown in Figure 3. Three threshold values are shown as three intensities in the picture (low, medium and high with the extra high omitted). The horizontal and vertical extrema have been combined.

B. Turning Texture to Gray Level

The next stage in our approach is to count the number of each size extrema in a window centered about each point. This results in a gray level picture representation of a texture property. For example, using a 40x40 window and three thresholds, the three pictures in Figs. 4, 5 and 6 were produced. Note that the forest region of the original has few small extrema, many medium extrema, and quite a few large extrema. The original image was also averaged using the same 40x40 window to produce a fourth picture, shown in Figure 7, representing the average gray level.

III. SEGMENTATION

We now have four pictures (or one 4-dimensional image) to be used for segmentation. Each picture element is considered to be a 4-dimensional vector. To accomplish segmentation, a starting point in each separate segment of the image is found. This is accomplished by finding local extrema in each of the four windowed pictures of Section II. In this operation a point must be a local extrema in both the horizontal and vertical directions to be chosen. This prevents the location of starting points in transitional

areas between two regions. The starting point candidates are then compared using a four-dimensional distance measure. Each group of similar candidates, based on a threshold criterion, are merged to produce an average vector representing that group. The resulting average vectors form the starting points for the segmentation. The distance measure used indicates an approximate percentage difference in each dimension:

$$D = \sum_{i=1}^{4} \frac{|A_i - B_i|}{A_i + B_i + K}$$

where A is the intensity of one point in the image and B is another. This measure is similar to gray level contrast. The constant K allows for decreasing the weight of a dimension in a region where the total number of extrema is small and, therefore the percentage difference is unreliable. For a window size of 40x40 we used a K=25.

Once the final set of starting points is determined, each point in the image, regardless of its spatial location, is assigned to the closest starting point using the distance measure described above. This normally results in fairly large contiguous regions due to the nature of the earlier windowing operations. Results using this technique on the image and intermediate steps presented in Figures 2 through 7 are shown in Figures 8 and 9. Figure 8 results when a large distance threshold criterion for similar starting point vectors is used. The additional region shown in Figure 9 was obtained by tightening this threshold. The major regions extracted from the original image are forest and two different grassy areas.

A simple biproduct of this segmentation is the region boundaries. A simple processing procedure on the segmentation output produces the boundary image shown in Figure 10. These boundaries are then shown overlaid on the original images in Figure 11.

IV. HIERARCHICAL SEGMENTATION

In order to completely segment an image, we propose to follow the above procedure using diminishing window sizes. The first large window size as described in the earlier sections will find major boundaries. The diminishing window sizes will find smaller objects which are averaged out by a large window. However, the smaller sizes will lose the more global context obtained by the larger ones and may miss some of the major boundaries.

To overcome this effect, each major region previously extracted will be subdivided separately. This prevents the proliferation of starting points and distance measurements. It also introduces spatial context into the segmentation procedure.

V. TECHNIQUE PARAMETER SENSITIVITY

There are several thresholds which must be set to make this technique operative: extrema sizes, window sizes, and distance similarity criteria. However, if the input data is fairly homogeneous (e.g., aerial photographs from a constant altitude) the algorithm performs well using fixed parameters. The algorithm is theoretically invariant to illumination level changes and magnification if the window sizes used are appropriate to the size regions to be detected.

VI. ALGORITHM IMPLEMENTATION

The one-dimensional extrema detection algorithm is easily implemented in a line-at-a-time digital processor. The picture is presently transposed and the process repeated to obtain the vertical extrema. It is feasible to implement a two-dimensional version of this algorithm using CCD transversal filter technology which would output extrema sequentially in real time and eliminate the time consuming transposition.

The smoothing operations described are implementable digitally, optically, or with CCD devices. Thus the overall segmentation system could be implemented for very fast image processing rates.

REFERENCES

1. A. Rosenfeld and A. C. Kak, Digital Picture Processing, Academic Press, New York, 1976. See especially Ch. 8 "Segmentation: for a discussion of present techniques and a list of references.

2. J. S. Weszka, C. R. Dyer, and A. Rosenfeld, "A Comparative STudy of Texture Measures for Terrain Classification," IEEE Trans. Syst., Man, Cyber., Vol. SMC-6, pp. 269-285, April 1976.

3. R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," IEEE Trans. Syst., Man, Cbyer., Vol. SMC-3, pp. 610-621, November 1973.

4. O. R. Mitchell, C. R. Myers, and W. Boyne, "A Max-Min Measure for Image Texture Analysis," IEEE Trans. on Computers, in press.

5. T. N. Cornsweet, Visual Perception, Academic Press, New York, 1970.

Figure 1 Sample gray level pattern for extrema detection. Local extrema of size 1 and size 3 are indicated.

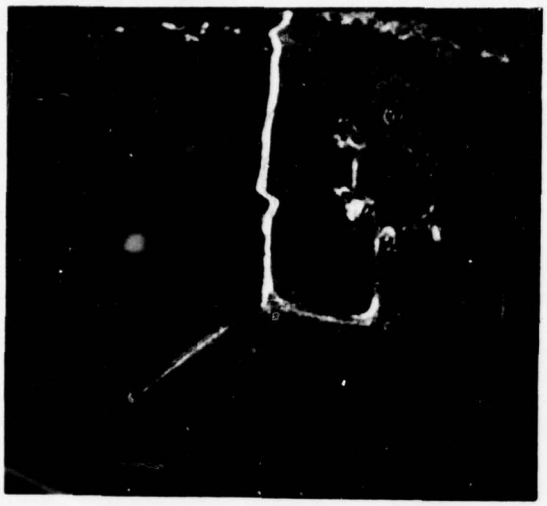


Fig. 2 Original image to be segmented, 256x256 8 bits per point

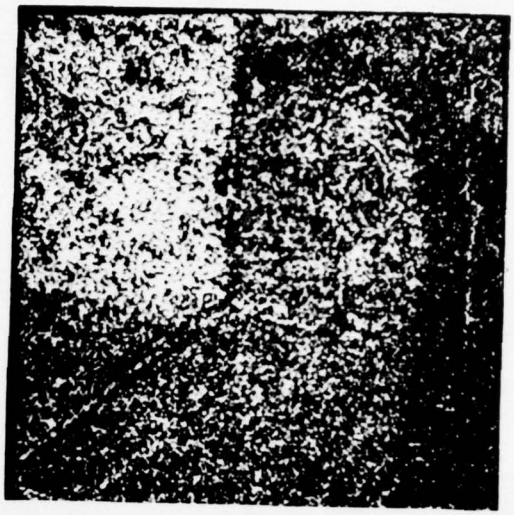


Fig. 3 Three levels of extrema displayed in different intensities

Fig. 4 Average low level extrema - each point represents the total number of low level extrema within a 40x40 window

Fig. 5 Average medium level extrema - each point represents the total number of medium level extrema within a 40x40 window centered at each point.

Fig. 6 Average high level extrema each point represents the total number of high level extrema within a 40x40 window centered at each point

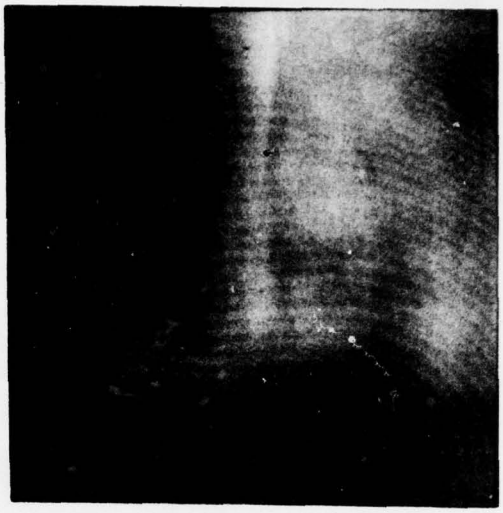


Fig. 7 Average gray level - each point represents the average gray level within a 40x40 window centered at each point

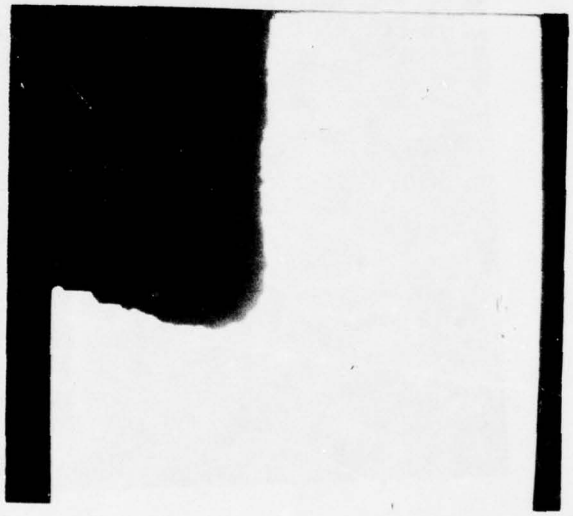


Fig. 8 Results of the segmentation procedure with loose threshold criterion on starting points

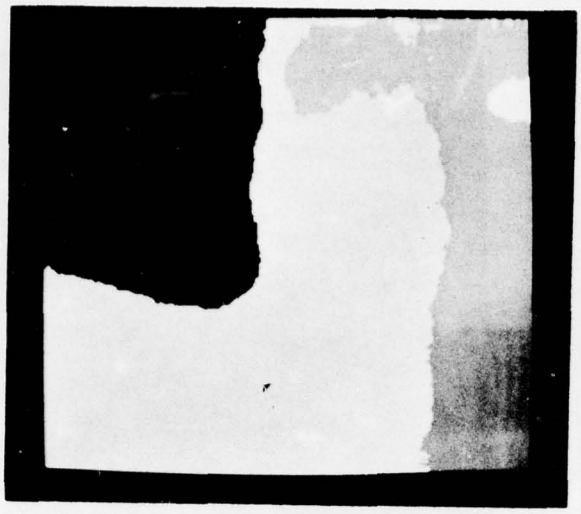


Fig. 9 Results of the segmentation procedure with a tight criterion on starting point generation

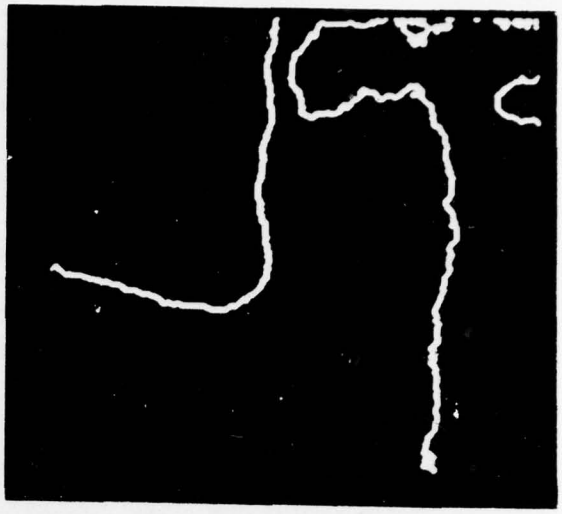


Fig. 10 Image boundaries produced from the segmentation output

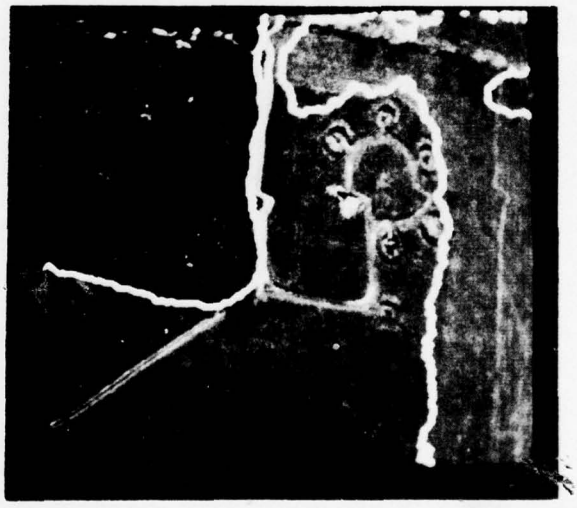


Fig. 11 Image boundaries outlaid on the original image

A MODEL OF AUTOMATIC INFORMATION EXTRACTION FOR IMAGE UNDERSTANDING

K. S. Fu and J. Keng

Models of information extraction are usually statistical models. Sometimes, in the implementation of these models, a man-machine interactive process is required to improve the processed results. Todd and Baumgardner [1] applied the statistical model which used spectral analysis for land-use classification of Marion County, Indiana (indianapolis area). Their results reidentified the problems that statistical analysis can not be applied to a fully satisfactory level. One of their difficulties is to distinguish highways and commercial areas by computer automation. With that difficulty in mind, we have proposed a syntax-directed method for automatic information extraction.[2]. This method for automatic information extraction consists of three major parts (see Fig. 1): preprocessor, syntactic analyzer, and postprocessor.

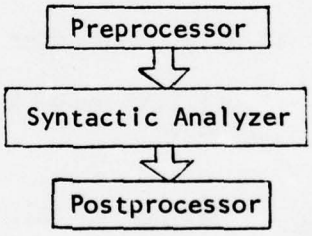


Figure 1 A model for automatic information extraction.

The preprocessor extracts primitives from an input image and transforms the image to "language" sentences which are the inputs for the syntactic analyzer. This process usually involves a transformation operation, thresholding operation, averaging operation and so forth. The syntactic analyzer processes the input image by a set of grammatical rules inferred by a grammatical inference

algorithm. This process is to accept those patterns which can be generated by the set of grammatical rules, and reject all others. The postprocessor sequentially executes several semantic rules. Those objects which are related, based on semantic information, are extracted from the image.

An example of urban development information extraction from LANDSAT image is given to illustrate the details of the method and to demonstrate its capability. First some special terms need to be defined.

A LANDSAT image is defined in the Euclidean space by the expression X(i,j,k) as

$$\{X(i,j,k) \mid 1 \leq i \leq N,\ 1 \leq j \leq M,\ 1 \leq k \leq 4\}$$

(N = the number of rows of the image)

(M = the number of columns of the image)

(k = the spectral band number)

The Thresholding Function is a Boolean function defined as

$$f_0^1 = Q(i,j,k) \begin{cases} 1 & \text{if } Q(i,j,k) \text{ greater than the threshold} \\ 0 & \text{if } Q(i,j,k) \text{ equal to or smaller than the threshold} \end{cases}$$

The Line Smoothing Process (LS) is defined in [2]. The Syntax-Directed Analyzer (SDA) is defined in [2]. The following is an implementation of this model for the extraction of information based on urban development.

1. Preprocessing

   Step 1 = Input LANDSAT images $\{X(i,j,k),\ 1 \leq i \leq N,\ 1 \leq j \leq M,\ 1 \leq k \leq 4\}$

Step 2 = Transformation Process $P(i,j,k') = X(i,j,1) + X(i,j,2)$

$P(i,j,k'') = X(i,j,3) + X(i,j,4)$

Step 3 = Thresholding Process $f_0^1(P(i,j,k'))$

$\overline{f_0^T}(P(i,j,k''))$

Step 4 = Line Smoothing Process $LS(f_0^1(P(i,j,k')))$

$LS(\overline{f_0^T}(P(i,j,k'')))$

2. Syntactic Analysis

HR and RR are Highway and River recognition results.

$HR(i,j,k') = SDA(LS(f_0^1(P(i,j,k'))))$

$RR(i,j,k'') = SDA(LS(\overline{f_0^T}(P(i,j,k''))))$

3. Postprocessing

Step 1 = $P(i,j,k') = P(i,j,k') \oplus HR(i,j,k')$

Step 2 = $P(i,j,k') = EX(P(i,j,k')$

EX is the execution process of the semantic information. For commercial/industrial the semantic rule is that if the area has over 60% of it to be concrete, then this area is called commercial/industrial area, since the commercial/industrial area has a high density of concrete buildings and concrete storage yard.

Step 3 = Region Growing Process

If the window $\{P(i+c,j+c,k) \mid 0 \leq c \leq 3\}$ satisfies step 3 then $P(i+c,j+c,k)$ is recognized as commercial/industrial area.

Step 4 = Achieving the urban development information as highway structures, river locations, and commercial/industrial expansion.

This method was implemented on the IBM 360/67 timesharing computer of the Laboratory for Applications of Remote Sensing in FORTRAN. The experiments have been conducted on different areas. Figure 2(a) is a satellite image (96 x 96 pixels) of a section within the northwestern part of Indianapolis area, taken in 1972. Figure 2(b) is the topographic map of the same area. This map was made in 1967. In Fig. 2(c) the intermediate output after the preprocessing is given. Figure 2(d) is the commercial/industrial area recognition result. This area is identified as the Lafayette Square Shopping Center near Indianapolis. Figure 2(e) is the urban development information extraction result. H, R, and C represent the highway, river, and commercial/industrial pixels, respectively. The interstate highway 65 (upper right part) goes into the city from the northwest. Highway 465 (from north to south) surrounds the city and highway 74 (left lower part) goes into the city from west. The Lafayette Square Shopping Center is located in the suburb and close to the highway. This area is typical of the urban development found in many large cities in the United States and this urban development information was automatically extracted by the proposed method.

Comparing Fig. 2(e) with the topographic map (1967) we can observe the growth of the commercial area clearly on the northern and eastern parts of

shopping center. This indicates that the automatic information extraction could be useful in topographic map making and updating.

Several large images (192 x 192 pixels) were also processed by the same technique. Figure 3(a) is the satellite image of Lafayette, Indiana area. Figure 3(b) is the commercial/industrial area recognition result. Figure 3(c) is the urban development information extraction result. Figure 3(d) is a city map of Lafayette. The commercial areas along the U.S. Highway 52 by-pass are successfully recognized. The Wabash river dividing West Lafayette and Lafayette is also recognized. Another example is provided in Figures 4(a)-(d). Figure 4(a) is a satellite image (192 x 192 pixels) of the northwestern part of Indianapolis, Indiana (Fig. 4(b)). Figure 4(c) is the commercial/industrial area recognition result. Figure 4(d) is the urban development information extraction result. The Lafayette Square Shopping Center and part of the downtown area (right lower part) are successfully extracted and recognized.

REFERENCES

1. W. J. Todd and M. F. Baumgardner, "Land-Use Classification of Marion County, Indiana, by Spectral Analysis and Digitized Satellite Data," LARS Information Note 101673, LARS, Purdue University, 1973.

2. J. Keng and K. S. Fu, "A Syntax-Directed Method for Land-Use Classification of LANDSAT Images," Proceedings of the Symposium on Current Mathematical Problems in Image Science, Monterey, CA, Nov. 10-12, 1976.

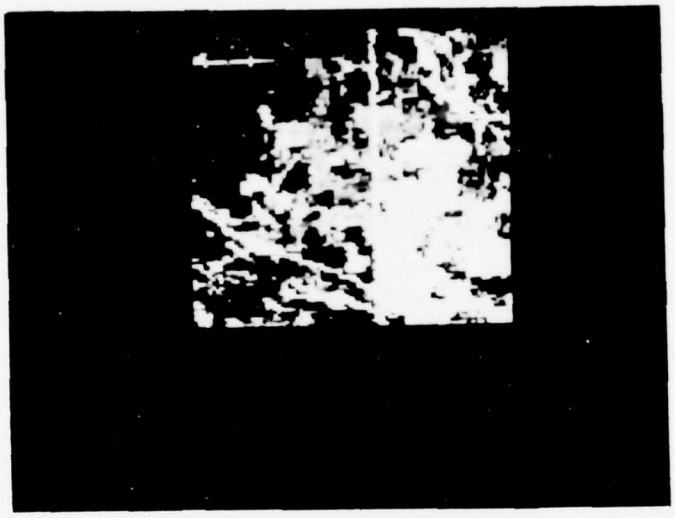


Figure 2(a) Satellite image of northwest part of Indianapolis, Indiana area (96 x 96 pixels).

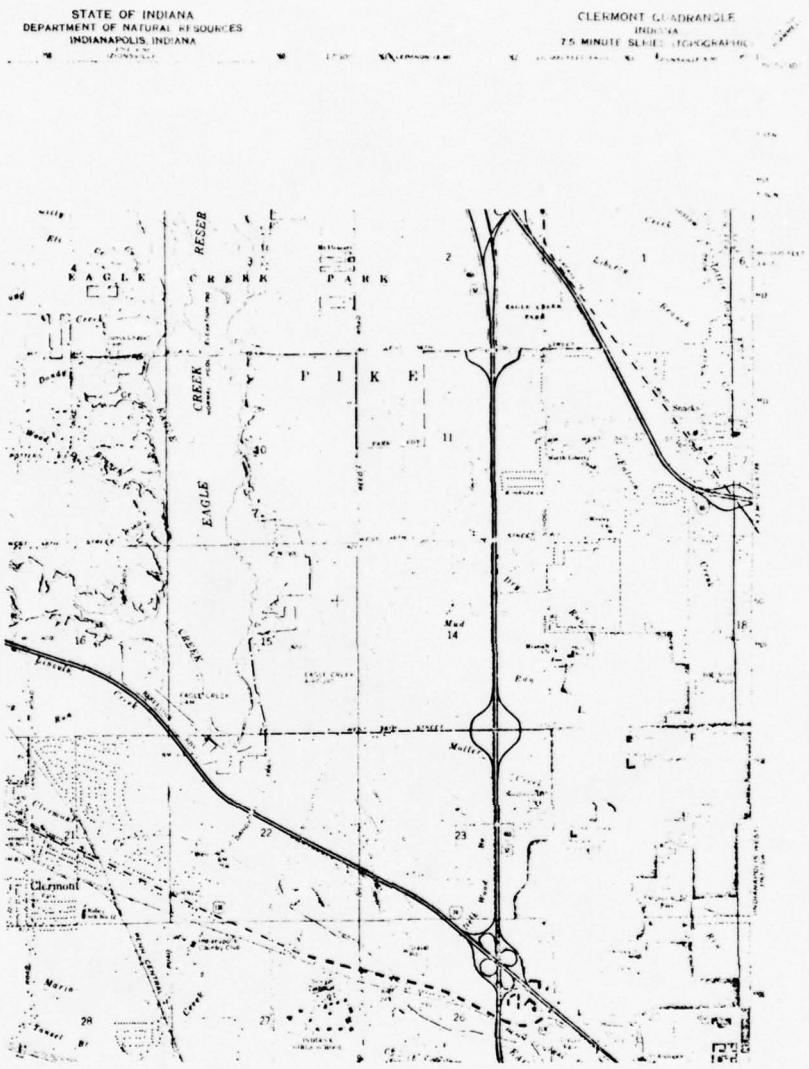


Figure 2(b) Topographic map of same area as Fig. 2(a).

Figure 2(c) Intermediate output after preprocess of Fig. 2(a).

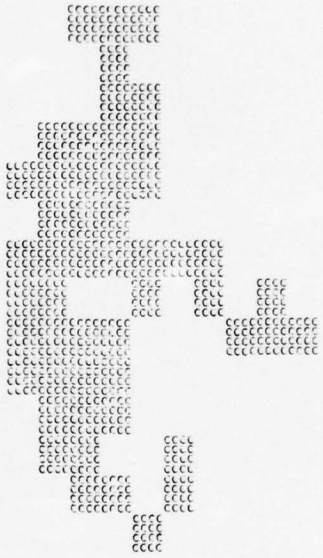


Figure 2(d) Commercial/industrial area recognition result.

Figure 2(e) Urban development information extraction result of Fig. 2(a).

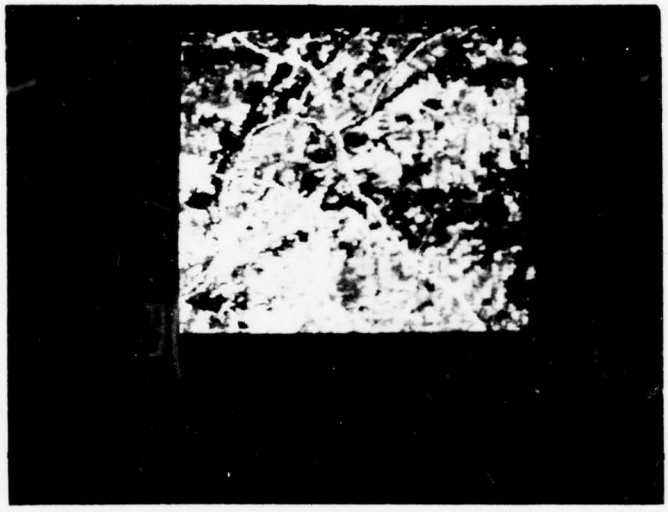


Figure 3(a) Satellite image of Lafayette area, Indiana (192 x 192 pixels).

Figure 3(b) Commercial/industrial area recognition result of Fig. 3(a).

Figure 3(c) Urban development information extraction result of Fig. 3(a).

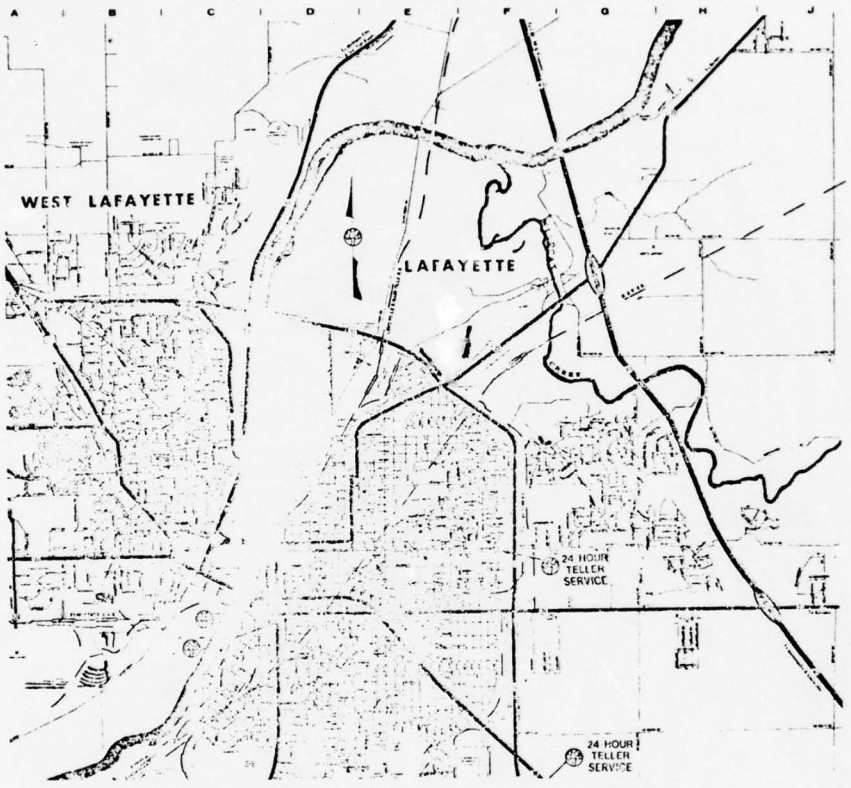


Figure 3(d) City map of Lafayette, Indiana.

Figure 4(a) Satellite image of northwest quarter of Indianapolis, Indiana (192 x 192 pixels).

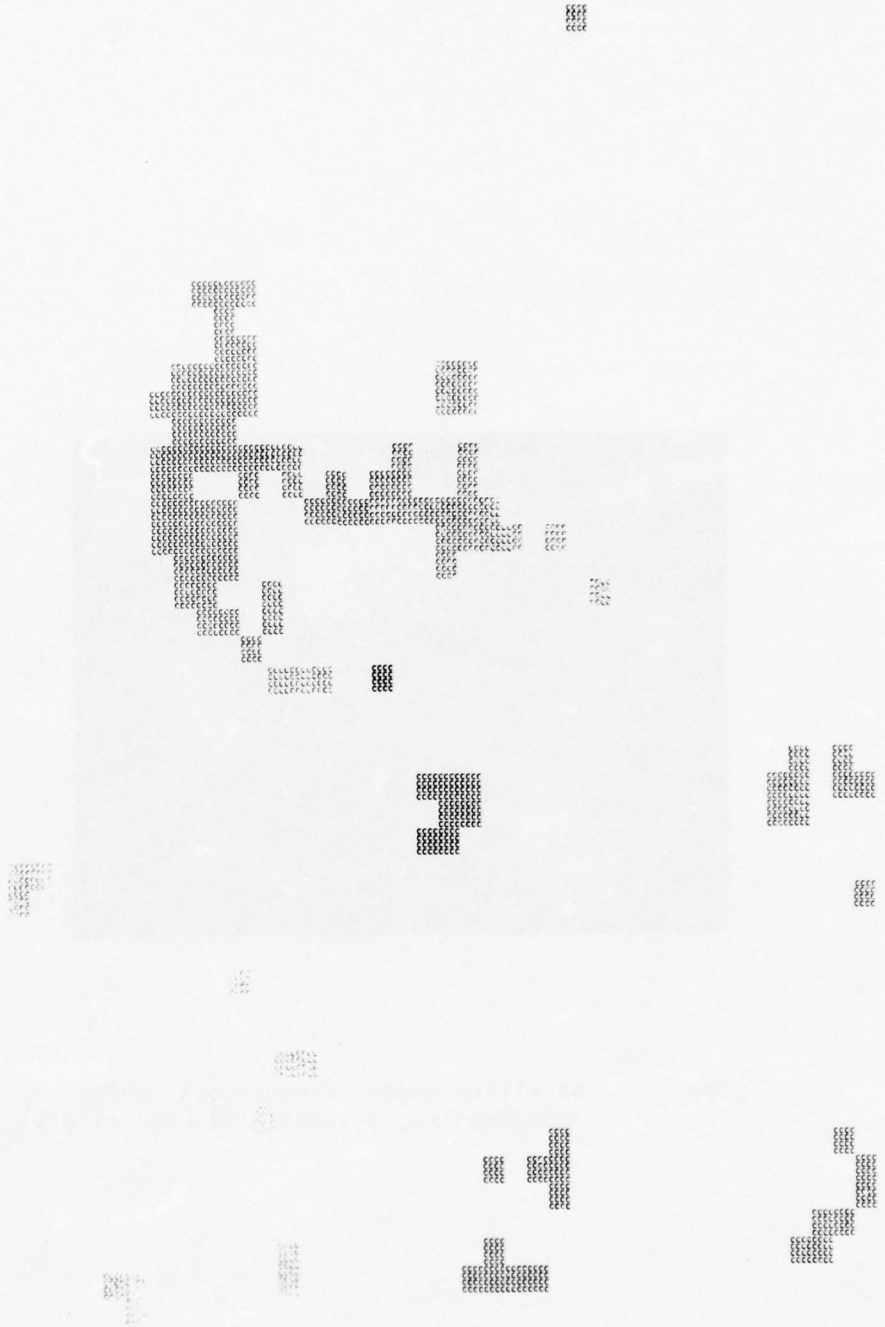


Figure 4(b) Commercial/industrial recognition result of Fig. 4(a).

Fig. 4(c) Urban development information extraction result of Fig. 4(a).

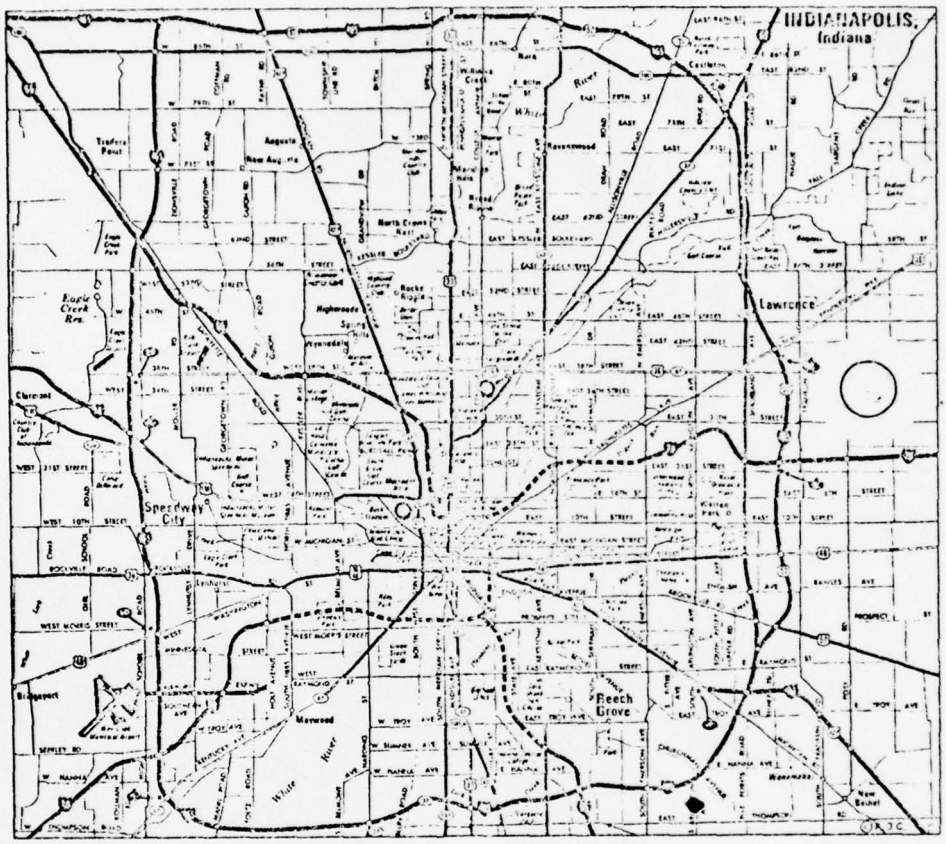


Figure 4(d) City map of Indianapolis, Indiana.

GENERALIZED CLUSTERING FOR PROBLEM LOCALIZATION

K. Fukunaga and R. D. Short

1. Introduction

In the past, applications of clustering have been limited to situations in which a set of data samples is given with no information as to its underlying density or class assignment. The clustering procedures are used to find natural divisions within the data set and thus learn something of the structure of the underlying density or a natural classification of the data [1]-[5].

For this paper we would like to introduce what we feel is a new application for clustering. In our case we assume that more is known about the sample space than just the samples themselves. Clustering is used to divide the space into disjoint regions, minimizing a criterion which takes into account the *additional* information. The resulting clusters are then used for the processing of new data samples.

The following sections describe the techniques involved in detail. Section 2 gives us introduction to the general approach with specific examples which apply to piecewise linear classification and piecewise linear density estimation. Section 3 describes in detail the algorithm for piecewise linear classifier design. Experiments involving the two class problem are given along with the results. In Section 4 we discuss the piecewise linear density estimation problems. Again experimental results are presented. Finally, the paper is concluded with a few remarks reflecting our impressions of the general procedure and the supporting experimental results.

2. Clustering for Problem Localization

For many problems in pattern recognition the researcher is seeking to find a global solution, that is a solution which is valid for the entire probability space. Quite often the solution can be greatly simplified if the space is first partitioned and the problem is solved in each local region. For example, in the classifier design problem the boundary between two classes may be rather complex in the space as a whole, while if the proper partitioning is chosen, linear classification in the subspaces may give excellent results. So if the correct subdivision of the space is found, a complex decision boundary may be reduced to a comparatively simple boundary, such as a linear one, in each region.

Another problem which is easier to solve at a local level is density estimation. Using Taylor's expansion of the density function

$$p(x) = p(x_o) + \nabla p(z)\Big|_{z=x_o}^{T} (x-x_o) + 0(||x-x_o||^2) \qquad (1)$$

we see that locally $p(x)$ can be estimated by a linear function

$$p(x) \simeq c_o + V^T x \ . \qquad (2)$$

If the researcher does opt for this method of solution he is immediately faced with the problem of finding the most desirable partitioning. The best partitioning for classifier design may give poor results when used for density estimation. The proper partitioning of the space will also depend on the type of classifier or density estimate to be used in each region. In order to find an optimum partitioning, we propose to use the general method of clustering with important modifications which are appropriate for the particular problem.

In conventional clustering we use a criterion of the form $J(X, \Omega)$, where X is the set of N, n dimensional data samples to be clustered and Ω is a partitioning of X [6]. Thus the only information used to perform the clustering is contained in the spatial coordinates of the data samples in X. For this reason the term 'unsupervised clustering' is often applied. If the data set gives a good representation of the underlying probability density, then various characteristics of the density function may be stressed in the clustering procedure. Such examples are the modal seeking and valley seeking properties of various criteria as found in the literature [2], [3]. Unfortunately, for our proposed purpose, the information available from the spatial coordinate values does not supply sufficient information for meaningful clustering. Therefore, we wish to introduce clustering criteria which take into account the specific application, we have in mind for the resulting partitioned space.

Consider the problem of a two class classifier design. Here we wish to divide the space into M regions. In each region we find the linear decision boundary which best classifies the data samples in that region. Since the ultimate goal for any classifier design is to minimize the probability of error, we choose for our clustering criterion the following

$$J(X, \Omega, V_1,\ldots,V_M, c_1,\ldots,c_M) = \sum_{i=1}^{M} \Pr\{\Gamma_i\} \Pr\{\text{error}/V_i, c_i, \Gamma_i\} \tag{3}$$

where $V_i^T x + c_i = 0$ is the best linear decision boundary for region Γ_i. For a finite set $X = \{x_1, x_2,\ldots,x_N\}$ and a decision rule

$$V_i^T x + c_i \gtrless 0 \begin{cases} x \in \omega_1 \\ x \in \omega_2 \end{cases} \tag{4}$$

we can estimate (3) by

$$J(X, \Omega, V_1,\ldots,V_N, c_1,\ldots,c_N) = \frac{1}{N} \sum_{i=1}^{M} \sum_{x_j \in \Gamma_i} d(x_j, V_i, c_i) \tag{5}$$

where

$$d(x_j, V_i, c_i) = \begin{cases} 0 & \text{if } h(x) \text{ gives a correct decision} \\ 1 & \text{otherwise.} \end{cases} \tag{6}$$

The property which distinguishes this criterion from the more traditional forms for clustering is that we must know apriori the classifications of the data set X. In this sense X can be viewed as a training set which we can use to find an appropriate subdivision of the space R^n and the subsequent linear classifiers. These results will then be used to classify unknown samples.

Two related problems immediately arise from using the criterion as state in (5), with a conventional iterative algorithm such as is used in ISODATA, [2]. As an example, consider the situation shown in Figure 1. If we start with the initial regions shown in Figure 1(A) and the resulting classifiers, then all points in class 1 to the right of the linear classifier in region 2 will be placed in region 1. Similarly, the samples in class 2 to the left of the linear classifier in region 2 will be placed in region 3. The problem that arises here is how to separate the points in the shaded areas into their respective regions. This raises the second problem of how to place a new unknown sample x into Γ_j or Γ_k, or which classifier to be used to classify x. Obviously, x does not have apriori information as to the correct classification of x. Hence, using the new clustering criterion with the traditional clustering procedures gives results which are of no real help.

To correct these flaws we must somehow introduce a structure or a mathematical form for our region boundaries which is independent of the apriori class information. As such a boundary structure we restrict our partitions to be linearly separable and minimize (5) over this class of partitionings. In this way we can describe the boundaries between the partitions by linear functions and thus assign any point in R^n to a region without the apriori knowledge of class assignment. At the same time we have minimized, as much as possible, the number of misclassified samples from the training set.

It is important to stress here that the choice of linear boundaries is an engineering one, for computational considerations. The basic philosophy is that we must some how restrict the class of partitionings such that a new sample can be assigned to a region without apriori knowledge of its classification. For this reason we have chosen to define a boundary structure which can be used without such apriori knowledge. Another technique is to restrict the class of partitionings such that the nearest neighbor of x_j is in the same region as x_j. Then the new sample will be placed in a region according to its nearest neighbor.

Now consider the problem of density estimation. For this problem we divide the space into M regions and find the best linear estimate of $p(x)$ in each region. We use the mean-square-error to measure the closeness of our estimate. The appropriate clustering criterion is then

$$J(X, \Omega, V_1,\ldots,V_N, c_1,\ldots,c_N)$$

$$= \sum_{i=1}^{M} \Pr(\Gamma_i)\, E\{|p(x)-V_i^T x - c_i|^2/\Gamma_i, V_i, c_i\} \qquad (7)$$

where $V_i^T x + c_i$ is the best linear estimate of $p(x)$ in Γ_i for mean-square-error. For the finite set $X = \{x_1,\ldots,x_N\}$ we have

$$J(X, \Omega, V_1,\ldots,V_M, c_1,\ldots,c_M)$$

$$= \frac{1}{N} \sum_{i=1}^{M} \sum_{x_j \epsilon \Gamma_i} |p(x_j) - V_i^T x_j - c_i|^2 \qquad (8)$$

As in the previously mentioned classification problem, we see that clustering X with the criterion in (8) requires apriori knowledge about the data samples in X. In this case the required information is $p(x_j)$, or at least an estimate of this quantity. Now, if we wish to apply the resulting density estimate to a new sample x, we are unable to use J for deciding which region to place x in without the apriori value of p(x). Taking the same approach as in the previous problem, we again restrict the class of partitionings to those which have linearly separable partitions.

In the following sections we present in more detail the two applications mentioned here for this new form of clustering. We would first like to re-emphasize that these are specific applications of a more general technique of finding optimum space partitionings for the purpose of problem localization. This technique should be considered in any situation in which the designer is faced with a problem that is more easily solved locally than globally. We will now give two specific examples.

3. Piecewise Linear Classification

In this section we introduce in more detail the concept of using a modified clustering procedure in designing a piecewise linear classifier. Again the basic philosophy to this approach is to divide the space into M regions and design a good linear classifier in each region. The regions are modified according to the criterion in (5) and new linear classifiers are then found for each new region. The procedure is repeated until significant improvement in the criterion ceases.

For our experiments we have chosen to divide the training set into 3 regions. The boundary structure for these partitions is restricted to be of the form.

$$
\begin{aligned}
x_j^T U_1 + a_1 \gtrless 0 &\begin{cases} x_j \in \Gamma_1 \\ x_j \in \Gamma_2 \end{cases} \\
x_j^T U_2 + a_2 \gtrless 0 &\begin{cases} x_j \in \Gamma_1 \\ x_j \in \Gamma_3 \end{cases} \\
x_j^T U_3 + a_3 \gtrless 0 &\begin{cases} x_j \in \Gamma_2 \\ x_j \in \Gamma_3 \end{cases}
\end{aligned}
\tag{9}
$$

where the $x_j^T U_i + a_i = 0$ form linear boundaries between two regions. A sample is placed into the region which receives a majority vote. In the case in which each region receives one vote, the linear boundary closest to the sample is found and the decision of that boundary is reversed. The parameter vector U_1 is found by

$$U_1 = (M_1 - M_2)/||M_1 - M_2|| \tag{10}$$

where $$M_k = \frac{1}{N_k} \sum_{x_j \in \Gamma_k} x_j$$

with N_k = number of samples in Γ_k.

The threshold a_1 is adjusted for otpimum division of Γ_1 and Γ_2. The other U_k's and a_k's are found analogously. The boundaries of (9) with U_i's of (10) are perpendicular to the mean-difference vectors between classes, which were used in ISODATA [2]. The only difference here is the introduction of thresholds a_i's. Our experimental results show that the proper selection of a_i's is very important to obtain reasonable region boundaries particularly when the numbers of samples in each regions are significantly different. Also, it is recommended that the data be transformed so as to make the covariance matrix of the mixture distribution an identity matrix. This normalization of the mixture covariance makes the clustering results coordinate-independent in general [7].

For the two class problem which is considered in the experiments, we find a linear classifier in each region. For this we choose the optimum solution to Fisher's criterion which is

$$V_i = (\Sigma_i^1 + \Sigma_i^2)^{-1} (M_i^1 - M_i^2) \tag{11}$$

where M_i^k is the sample class k mean in region Γ_i and Σ_i^k is the sample class k covariance in region Γ_i. The threshold c_i is adjusted for optimum classification. We note that in both the case of the linear boundaries and the case of the linear classifiers we have avoided any iterative techniques which would be required for finding the optimum parameters. We feel that the computational savings of avoiding a two layered iterative algorithm well out weighs the loss in performance. From the experimental results, we have concluded that this decision is justifiable.

The general algorithm used in our experiments is as follows.

1) Choose an initial clustering of X, $\tilde{\Gamma}(0)$.

2) Find the U_i's from (10) and the a_i's as discussed above for $\tilde{\Omega}(K)$; the resulting partitioning is $\tilde{\Omega}(K)$.

3) Calculate the V_i's from (11) and the c_i's as described above.

4) Reassign all $x_j \in X$ according to the rule:

   a) Leave x_j in its present region if the linear classifier corresponding to that region correctly classifies x_j.

   b) Otherwise place x_j in the region corresponding to the classifier that most strongly favors the correct class. For example, in the two class problem, if the V_i's are normalized, we place x_j in region Γ_i corresponding to the maximum (minimum) of

$$x_j^T V_i + c_i \ ,$$

   for x_j in class 1 (class 2). The decision rule in Γ_i is given by

$$x_j^T V_i + c_i \gtrless 0 \quad \begin{cases} x_j \in \text{class 1} \\ \\ x_j \in \text{class 2} \end{cases}$$

   The new paritioning is denoted $\tilde{\Omega}(K+1)$.

5) If no x_j is reassigned, stop, otherwise increment K by 1 and return to step 2.

Clearly, with slight modifications this algorithm could be applied to the multiclass problem, using a multiclass linear classifier in each region. For example, if a majority vote type classifier is used in each region we would reassign x_j to the region which gives the correct class the most votes, if it were misclassified in its present region.

Experimental Results

Experiment 1

For this experiment 80 two dimensional samples are used to train a two class classifier with 40 samples in each class. Class 1 is normally distributed with zero mean and an identity covariance. Class 2 is equally divided between a normal distribution with a mean of (2.5,0) and a normal distribution with a mean of (-2.5,0), both having an identity covariance. Thus class 2 lies on both sides of class 1.

As in all experiments the data is first transformed in order to whiten the covariance of the mixture distribution, and all illustrations are shown in the transformed space. The initial regions and the final regions and corresponding classifiers are shown in Figures 2(A) and (B) respectively. The final results were found after 7 iterations. The classification results as applied to 320 test samples (independent of design samples) was 16.5 per cent misclassified. This compares to a theoretical Bayes error of 15 percent.

Experiment 2

Next we consider another two dimensional two class classification problem with class distributions as shown in Figure 3. Again 40 samples are used from each class for training of the classifier. After 6 iterations the original regions shown in Figure 3(A) were changed to the final regions and classifiers in Figure 3(B). It is interesting to note here the role that the region boundaries play in the complete decision boundary.

Experiment 3

For the final classifier design experiment we have chosen an eight dimensional two class problem. The training set consists of 320 samples, 160 from each class. Class 1 is normally distributed with mean and

covariance as identified with Standard Data 1 in [6]. The data in class 2 is equally distributed between normal distributions described by Standard Data 2 and Standard Data 3 in [6]. The values of the criterion in (5) are plotted in Figure 4. These results compare to an error of approximately 4.6 percent if the optimal linear boundary between class 1 and Standard Data 2 and the optimal linear boundary between class 1 and Standard Data 3 are used as the decision boundary [6].

4. Density Estimation

We would next like to discuss further the application of our clustering procedure to density estimation. Here we have chosen to divide the space into three regions and to find the best linear estimate of $p(x)$ in each region. The criterion for a given training set X is again

$$J(x_1,\ldots,x_N, V_1,\ldots,V_3, c_1,\ldots,c_3) = \frac{1}{N} \sum_{i=1}^{3} \sum_{x_j \epsilon \Gamma_i} |p(x_j) - x_j^T V_i - c_i|^2 . \tag{12}$$

This criterion implies that we must know $p(x_j)$ for all samples in the training set. Therefore unless we know the functional form of $p(x)$ we must use an estimate. Examples of such an estimate are the k-nearest neighbor and the Parzen estimate. For our experiment we use the n-dimensional Parzen estimate [8]

$$\hat{p}(x_j) = \frac{1}{N} \sum_{i=1}^{N} (1/h)^n \, k((x_j - x_i)/h) \tag{13}$$

where the kernel function is

$$k(x) = \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2} x^T x} \tag{14}$$

The selection of h is always crucial in the Parzen density estimate. However, the optimization of h is not our main concern in this paper, we selected $h = N^{-\frac{1}{n+4}}$ for n dimensional data with N samples [4], [8], [9].

As in the classifier design we restrict the regions to be separable by the linear boundaries described in (9) and (10) of the previous selection. The linear density estimates, $x^T V_i + c_i$, are optimized in each region to minimize the criterion

$$J_i = \frac{1}{N_i} \sum_{x_j \epsilon \Gamma_i} |\hat{p}(x_j) - x_j^T V_i - c_i|^2. \tag{15}$$

Taking the gradient of J_i with respect to V_i and the partial of J_i with respect to c_i gives

$$\nabla_{V_i} J_j = -\frac{2}{N_i} \sum_{x_j \epsilon \Gamma_i} (\hat{p}(x_j) - x_j^T V_i - c_i)\, x_j \tag{16}$$

and

$$\frac{\partial J_i}{\partial c_i} = -\frac{2}{N_i} \sum_{x_j \epsilon \Gamma_i} (\hat{p}(x_j) - x_j^T V_i - c_i)\ . \tag{17}$$

Setting (16) and (17) to zero and solving for V_i and c_i gives

$$V_i = \Sigma_i^{-1} \frac{1}{N_i} \sum_{x_j \epsilon \Gamma_i} (x_j - M_i)\, \hat{p}(x_j) \tag{18}$$

and

$$c_i = \frac{1}{N_i} \sum_{x_j \epsilon \Gamma_i} \hat{p}(x_j) - M_i^T V_i\ , \tag{19}$$

where M_i is the sample mean in Γ_i and

$$\Sigma_i = \frac{1}{N_i} \sum_{x_j \epsilon \Gamma_i} x_j\, x_j^T - M_i\, M_i^T\ . \tag{20}$$

Equations (18) and (19) give the optimum linear estimate of the density function $\hat{p}(x_j)$ in each region.

The following algorithm gives the general scheme used for the piecewise linear density estimation.

1) Choose an initial clustering $\tilde{\Omega}(0)$
2) Find the U_i's using (10) and the a_i's as mentioned in the previous section for $\tilde{\Omega}(K)$; the resulting partitioning is $\Omega(K)$.
3) Calculate the V_i's and the c_i's as in (18) and (19).
4) Reassign x_j according to the minimum

$$|\hat{p}(x_j) - x_j^T V_i - c_i|$$

to Γ_i for each x_j in X. The resulting partitioning is $\tilde{\Omega}(K+1)$.

5) If no x_j's reassigned, stop, otherwise increment K by 1 and return to step 2.

Notice that the only difference between this algorithm and the one in the previous section is in steps 3 and 4. These steps are dependent upon the particular application of the clustering and the criterion one chooses.

Experimental Results

Experiment 4

The above procedure was used to estimate a two dimensional data set with 160 samples. The data samples were generated with a normal distribution with zero mean and an identity covariance. The initial and final regions are shown in Figure 5. A plot of the criterion J is shown in Figure 6 which shows a convergence after 6 iterations.

Experiment 5

One hundred sixty were generated from an 8-dimensional normal distribution with zero mean and identity covariance. Convergence of the algorithm was similar to the two dimensional case.

5. Conclusions

From the experimental results we feel optimistic that the general approach which we call supervised clustering is indeed a viable method of region selection in problems such as piecewise linear classification and density estimation. This method may prove fruitful in a wide class of problems in which one needs to divide the probability space into local regions and thus attack the problem at a localized level. Another advantage to this technique is its easy adaptability to a tree structure for region division. That is, upon dividing the space into M regions we can apply this algorithm to each resulting region and thus further subdivide these regions into M subregions. This process can be repeated forming a tree structure with the desired number of levels. However it was found in our experiments that satisfactory results were obtained by using only a single division into three regions. Three regions were chosen primarily due to computational considerations.

It is particularly worth noting again that in the classifier design we were able to use relatively simple techniques for region boundary and linear classifier design. Very promising results were found using these techniques which enabled us to preserve a single level iteration algorithm. This results in considerable computational savings. For these reasons we feel that the results indicate that this method is competitive with other known approaches to classifier design.

References

[1] G. H. Ball, "Data analysis in the social sciences: What about the details?" in 1965 Fall Joint Comput. Conf., AFIPS Conf. Proc., Vol. 27, Moutvale, N.J.: AFIPS Press, 1965, pp. 533-559.

[2] G. H. Ball and D. J. Hall, "ISODATA - a novel method of data analysis and pattern classification," Technical Report, SRI Project 5533, Stanford Research Institute, Menlo Park, Calif., May 1965.

[3] W. L. G. Koontz and K. Fukunaga, "A nonparametric valley-seeking technique for cluster analysis," IEEE Trans. Comput., Vol. C-21, pp. 171-178, Feb. 1972.

[4] W. L. G. Koontz and K. Fukunaga, "Asymptotic analysis of a non-parametric clustering technique," IEEE Trans. Comput., Vol. C-21, pp. 967-974, Sept. 1972.

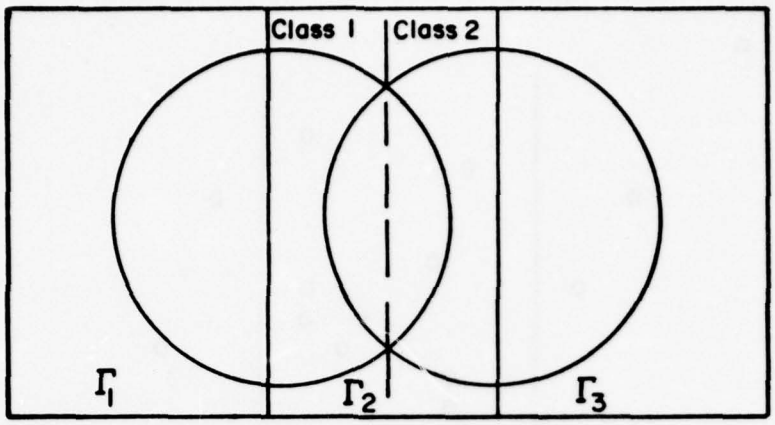
[5] M. S. Watanabe, Knowing and Guessing, New York: Wiley, 1969, Ch. 8.

[6] K. Fukunaga, Introduction to Statistical Pattern Recognition, New York: Academic Press, 1972.

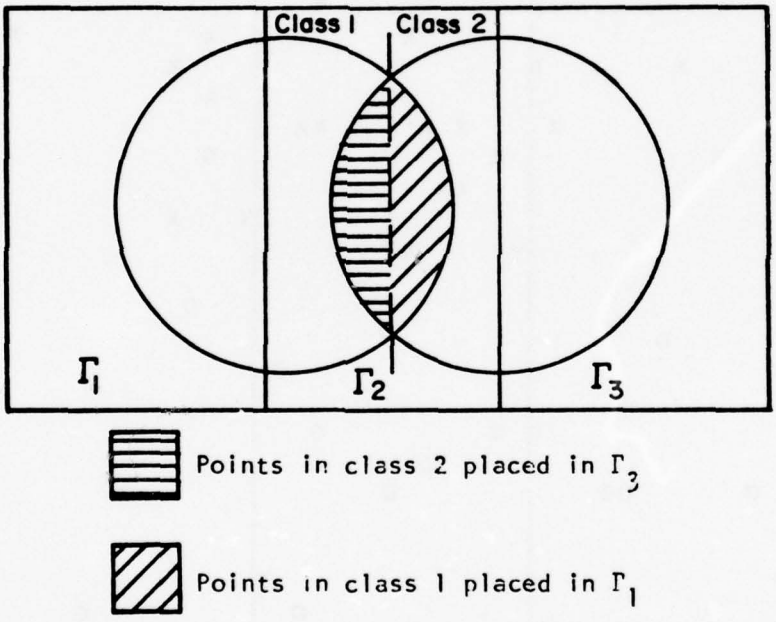
[7] K. Fukunaga and W. L. G. Koontz, "A criterion and algorithm for grouping data," IEEE Trans. Comput., Vol. C-19, pp. 917-923, Oct. 1970.

[8] E. Parzen, "On estimate of a probability density function and a mode," Ann. Math. Stat., Vol. 33, pp. 1065-1076, Sept. 1972.

[9] K. Fukunaga and L. O. Hostetler, "Optimization of k-nearest neighbor density estimates," IEEE Trans. Inf. Theory, Vol. IT-19, pp. 320-326, May 1973.



(A)



(B)

Figure 1: An example of clustering using criterion (3) with a conventional clustering algorithm.

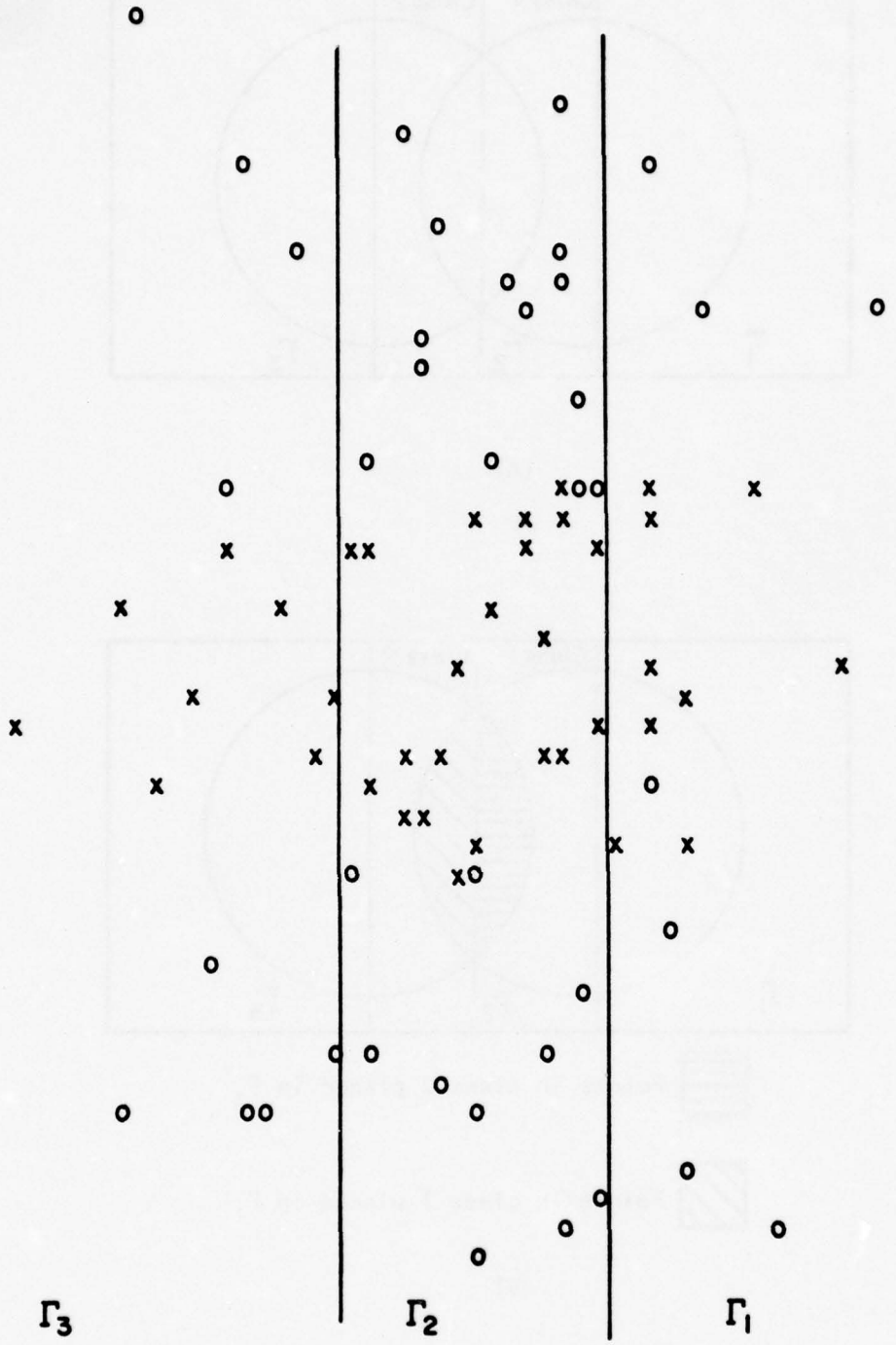


Figure 2A: Results of classifier design in experiment 1.

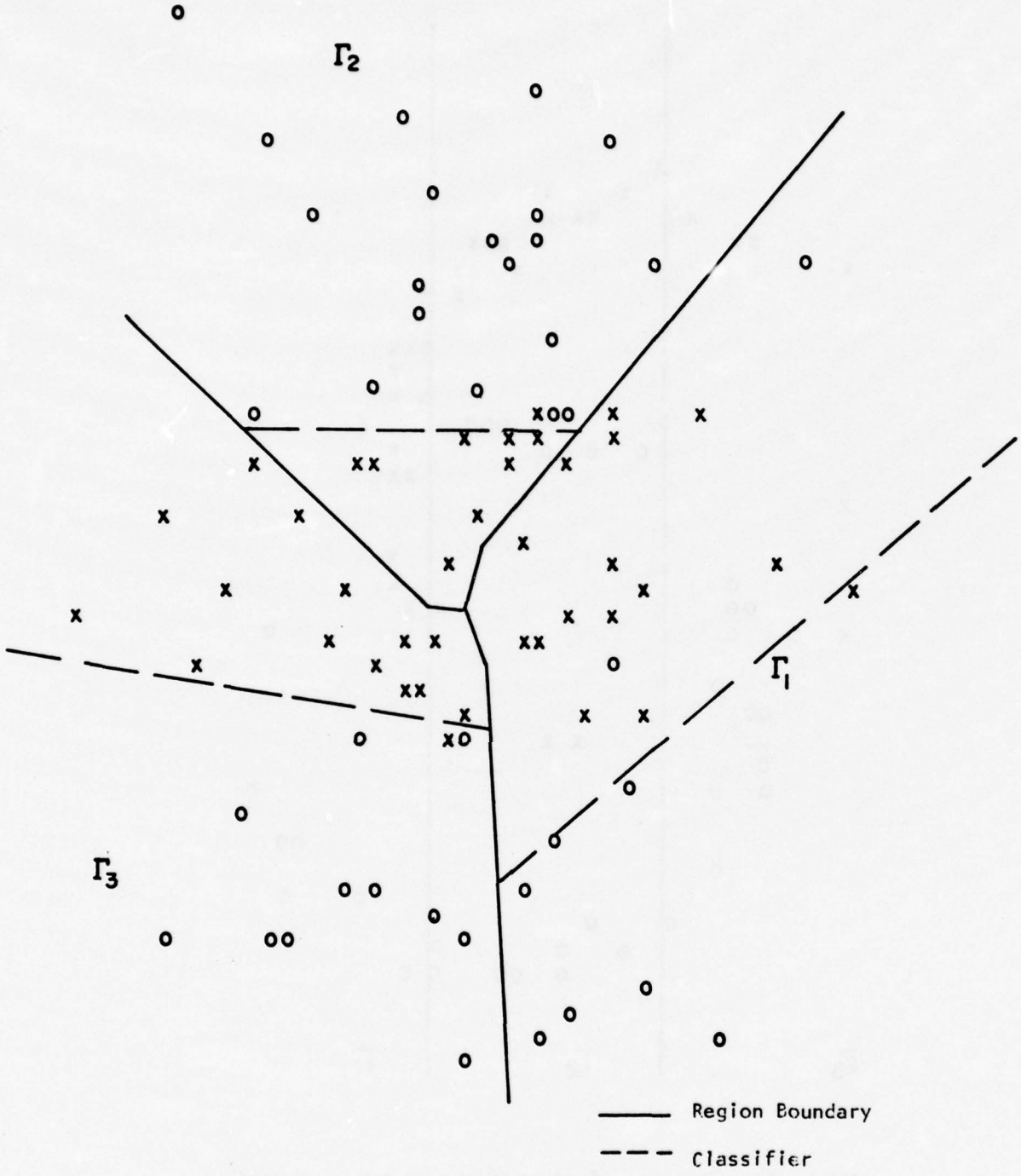


Figure 2B: Results of classifier design in experiment 1.

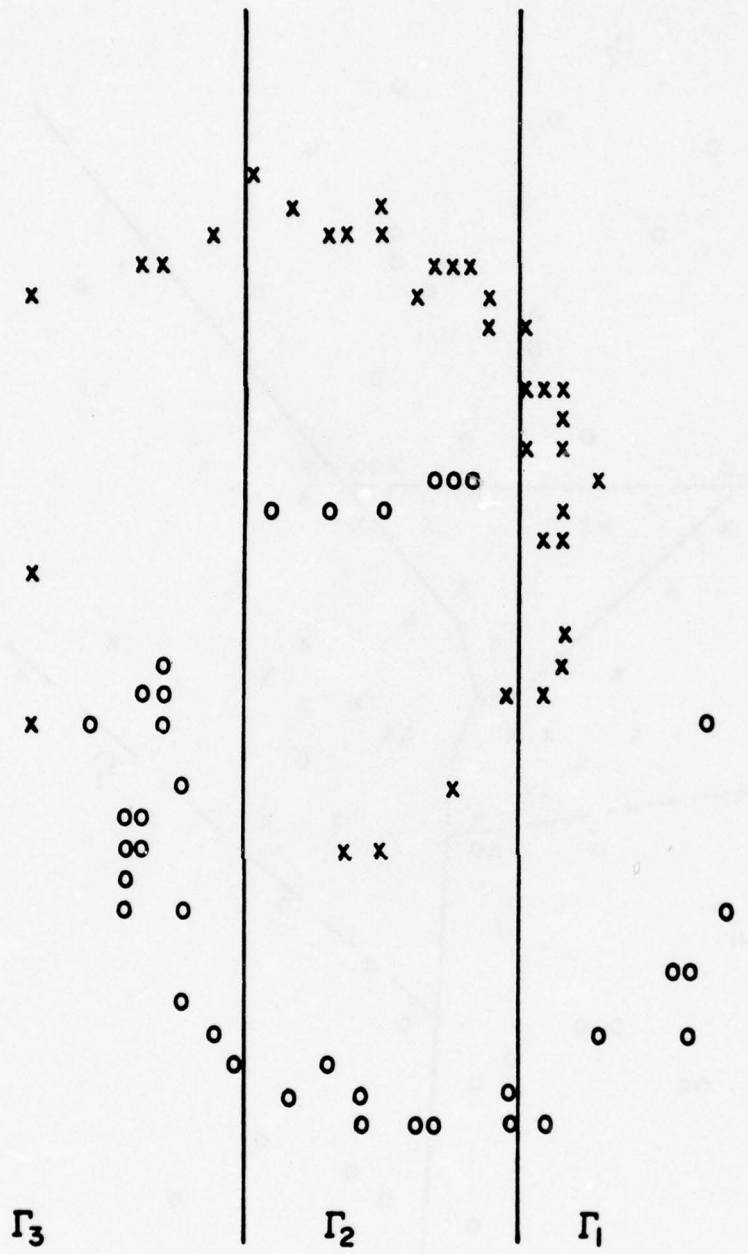


Figure 3A: Results of classifier design in experiment 2.

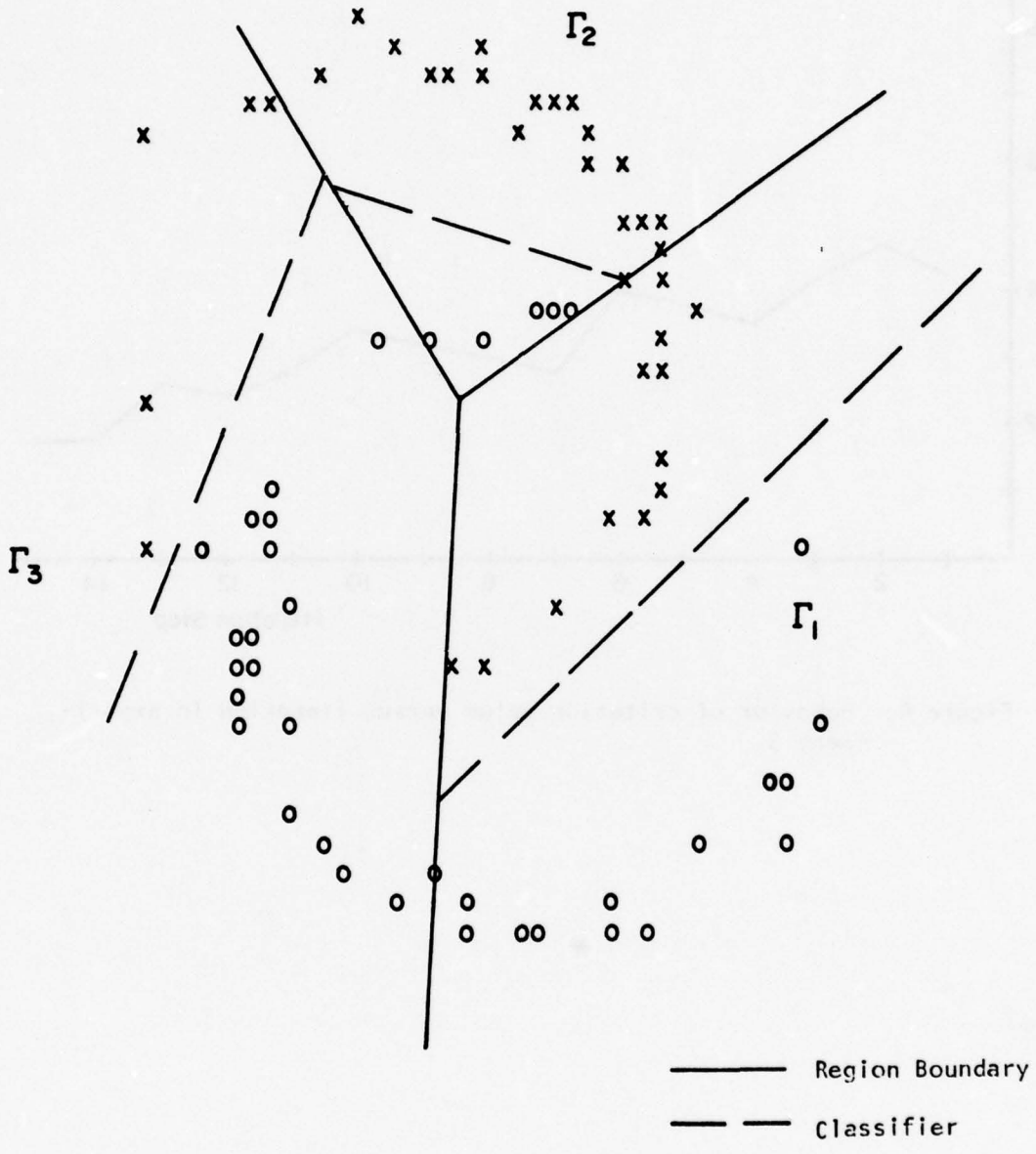


Figure 3B: Results of classifier design in experiment 2.

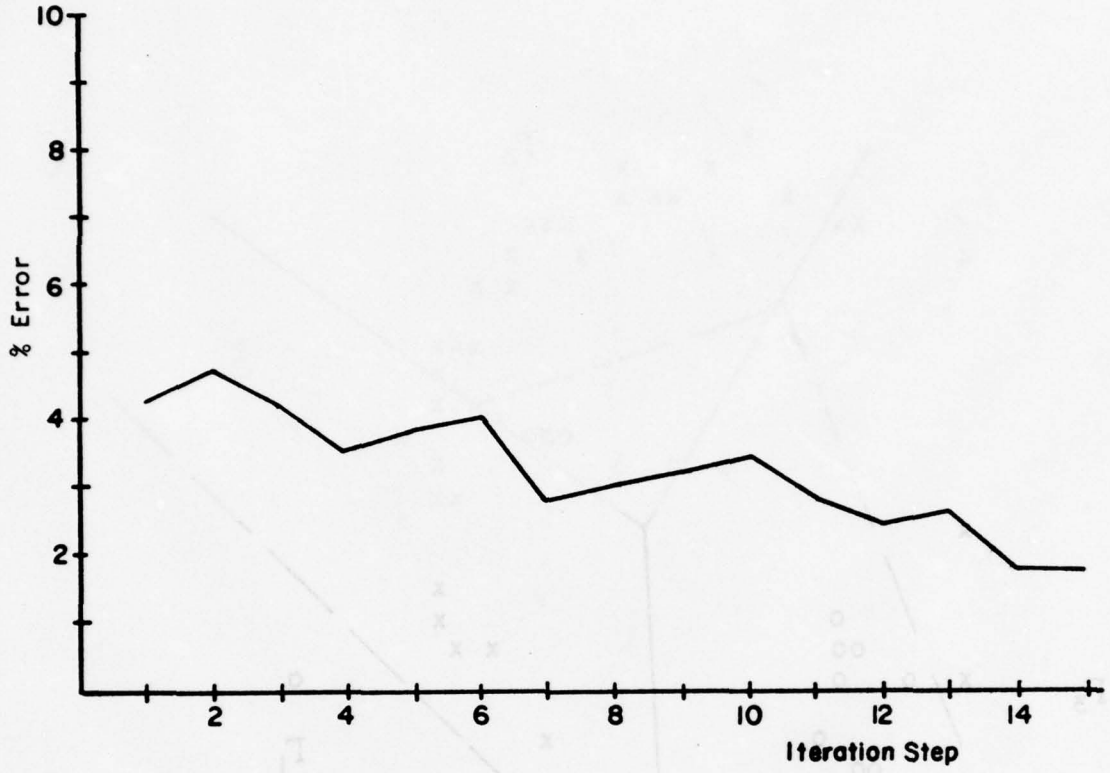


Figure 4: Behavior of criterion value versus iteration in experiment 3.

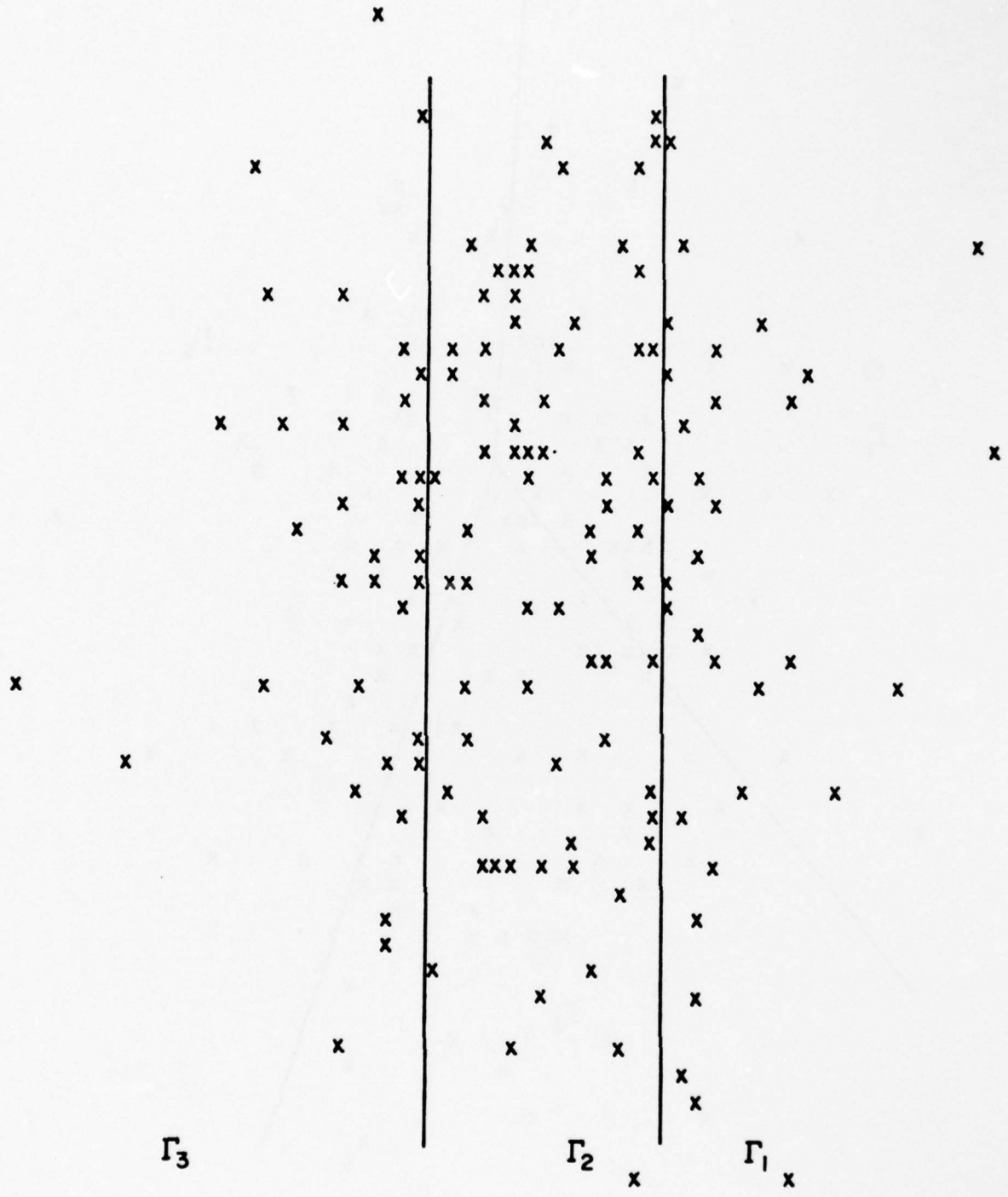


Figure 5A: Initial and final partitioning of space for density estimation in experiment 4.

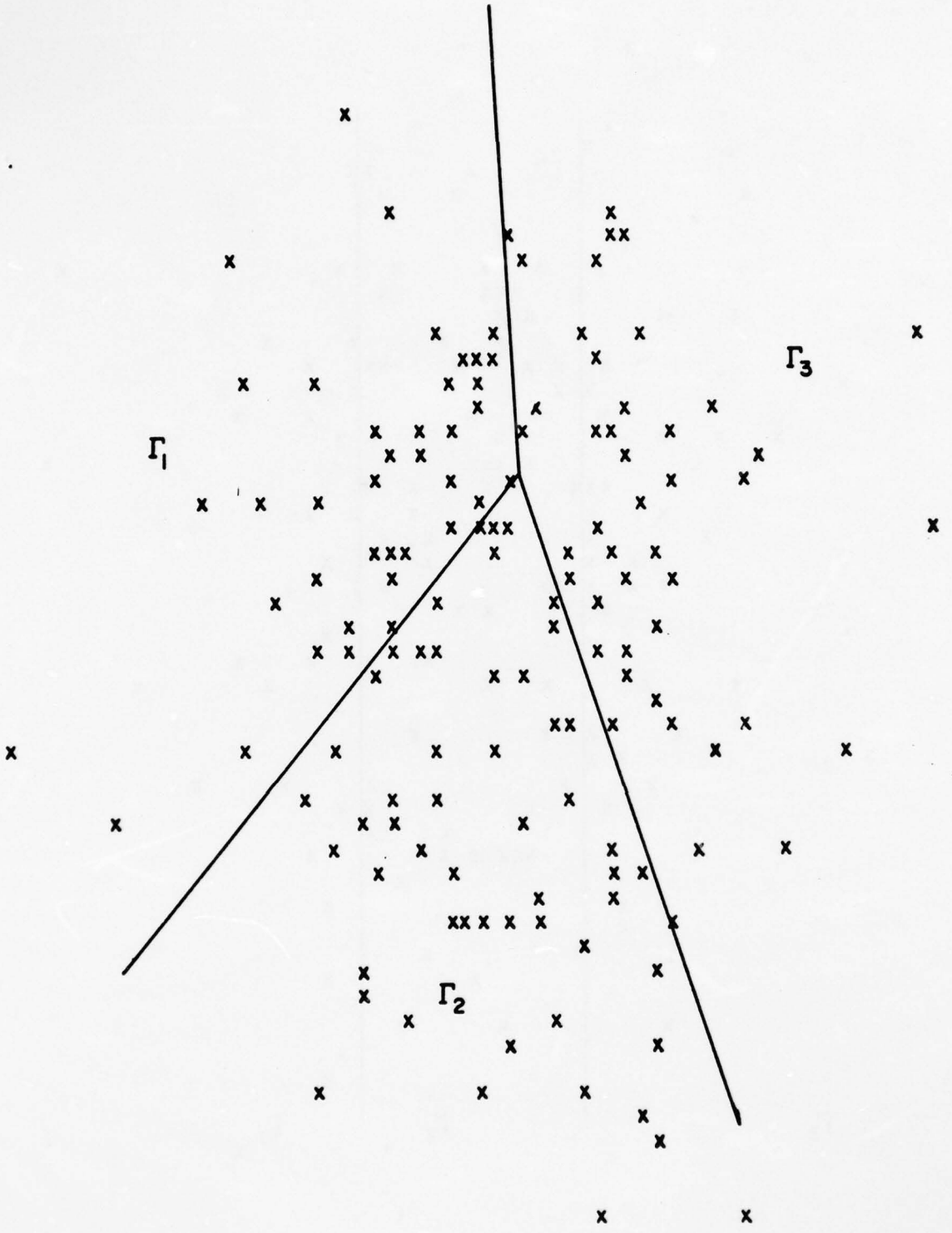


Figure 5B: Initial and final partitioning of space for density estimation in experiment 4.

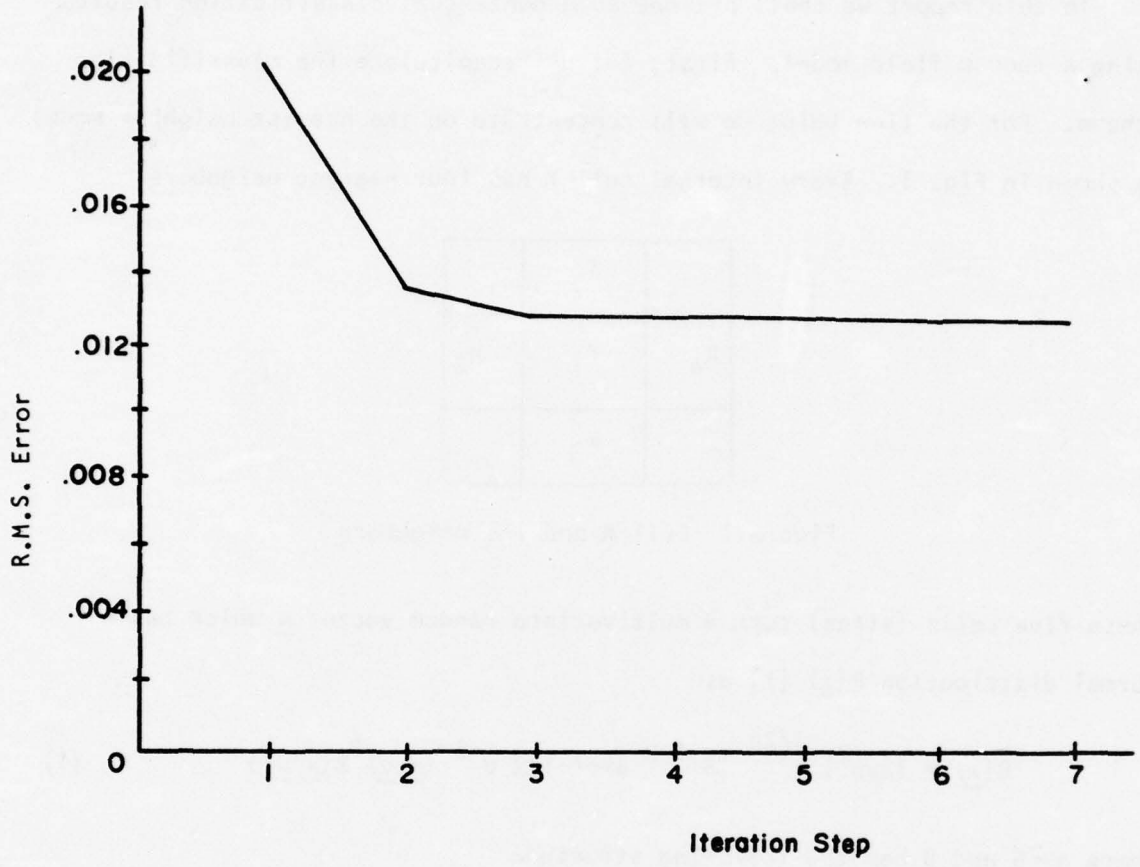


Figure 6: Mean square error versus iteration step for density estimation in experiment 4.

RANDOM FIELD APPROACH FOR STATISTICAL CLASSIFICATION USING CONTEXTUAL INFORMATION

K. S. Fu and T. S. Yu

In this report we shall present some contextual classification results using a random field model. First, let us recapitulate the classification scheme. For the time being we will concentrate on the nearest neighbor model as shown in Fig. 1. Every internal cell K has four nearest neighbors.

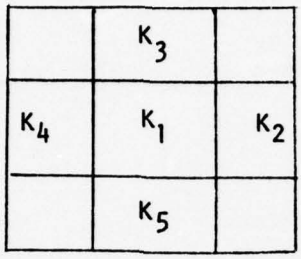


Figure 1 Cell K and its neighbors

These five cells (sites) form a multivariate random vector $\underline{x}$ which has a normal distribution $P(\underline{x})$ [1] as

$$P(\underline{x}) = (2\pi\sigma^2)^{-1/2n} |B|^{1/2} \exp\{-1/2\ \sigma^{-2}\ (\underline{x}-\underline{u})^T B(\underline{x}-\underline{u})\} \qquad (1)$$

where $n = 5$ and $\underline{B}$ has the following structure

$$B = \begin{bmatrix} 1 & -\beta_{12} & -\beta_{13} & -\beta_{14} & -\beta_{15} \\ -P_{12} & 1 & 0 & 0 & 0 \\ -\beta_{13} & 0 & 1 & 0 & 0 \\ -\beta_{14} & 0 & 0 & 1 & 0 \\ -\beta_{15} & 0 & 0 & 0 & 1 \end{bmatrix}$$

and σ^2 is the conditional variance of the conditional distribution:

$$P(x_k/x_{k_1},\ldots,x_{k_4}) = (2\pi\sigma^2)^{-1/2} \exp[-1/2\,\sigma^{-2}\{x_k-u_k - \Sigma\,\beta_{ij}\,(x_j-u_j)\}^2] \tag{2}$$

u_k is the mean value of random variable x_k. The proposed classification procedure is to choose the joint class $\underline{\theta}_k = (\theta_k,\theta_{k_1},\theta_{k_2},\theta_{k_4},\theta_{k_5})$ which minimizes the risk

$$R(L,\underline{\theta}_k) = \sum_{\underline{\theta}_\ell} L(\underline{\theta}_k,\underline{\theta}_\ell)\,P(\underline{x}/\theta_k)G(\underline{\theta}_k) \tag{3}$$

where G is the a priori distribution for $\underline{\theta}_k$. The classification of cell K is then a result of its own appearance as well as the appearance of its neighbors. Context is then used in classifying every cell. This algorithm is used to classify one set of satellite (LANDSAT) data.

The multispectral LANDSAT data used was provided by LARS (Lab for Application of Remote Sensing). The data covers an area of Lafayette, IN under run No. 72053609. The ground truth information was obtained by professional analysts. It was found that there are seventeen subclasses in the data set. Training of these classes were performed using approximately 3500 samples. A different area of size 128 x 128 was used to test the performance of the classification algorithm. The area covers lines (200,327), columns (120,247). If there is one-channel measurement associated with each cell, the random field model gives 50% classification accuracy while the corresponding simple (no context) decision gives 30% classification accuracy.

In order to make use of the multispectral properties of LANDSAT data, we employed a rather heuristic extension of the univariate case. The additional variates were treated as notational sites and the univariate model can be applied. The multivariate site-variable case is being investigated.

The two-channel classification gives 50% classification accuracy while the corresponding two-band simple classification gives 51% accuracy. The context algorithm is found to take five times longer in computer time than the simple classification algorithm. Results of classification together with ground truth are shown in Fig. 2. The results demonstrated the feasibility of using contextual information in classification. Multivariate site-variable case is of particular interest and is currently being investigated.

REFERENCES

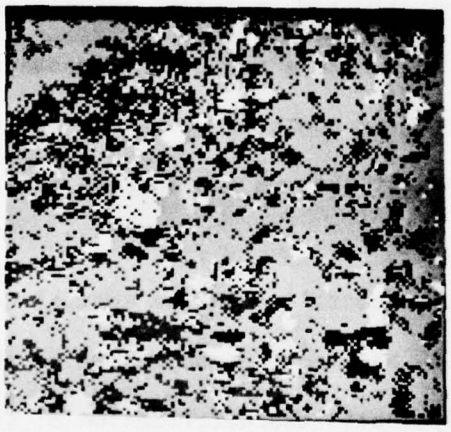
1. J. E. Besag, "Spatial Interaction and the Statistical Analysis of Lattice Systems," J. R. Stat. Soc. B, 36, pp. 192-236, 1974.

2. T. K. Cary and J. C. Lindenlaub, "A Case Study Using LARSYS for Analysis of LANDSAT Data," LARS Information Note 050575, LARS, Purdue University, West Lafayette, IN.

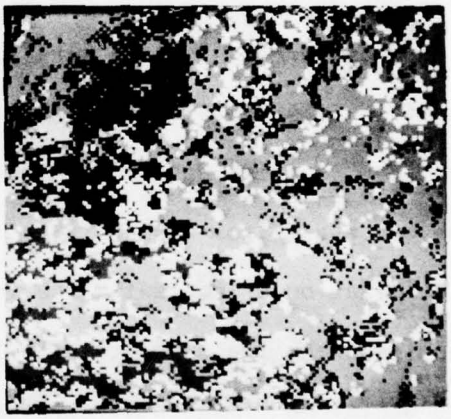
3. T. W. Anderson, Introduction to Multivariate Statistical Analysis, New York, John Wiley, 1958.

(a) Ground truth

(b) 1-band simple rule



(c) 1-band 4 neighbor rule



(d) 2-band 4 neighbor rule

Figure 2 Random field model results

MODELS FOR CLASSIFICATION USING SPATIAL AND TEMPORAL CONTEXT

P.H. Swain and W. Pfaff

Classifiers operating on multispectral imagery generally observe and classify a single image point at a time. Recent studies, however, confirm what our intuition and any good photointerpreter could tell us, that there is a great deal of information in the context in which an image point is found which can be helpful in identifying the point. Statements like, "Trucks are more likely to occur on roadways than in agricultural fields," describe the kind of context information which can help us discriminate trucks from tractors even though the resolution of the imagery is not adequate for us to discern the shape of the vehicle. Similarly: "A pattern of strips of asphalt and/or concrete in a non-urban area, limited in length and not connecting urban areas, is more likely an airport than a network of roadways." Thus, if we consider an object together with its context, we may be able to "understand" a great deal more about the object than we can when we consider the object apart from its context.

Utilization of context - the information contained in the spatial dependencies among image points - is therefore an important objective on the way to achieving "image understanding." Syntactic characterization of scene structure, described elsewhere in this report, is one way of utilizing context. Another is to think of a scene as being generated by a multi-dimensional random process characterizable in terms of its statistical transition properties.

A relatively simple approach to the statistical treatment of context in remote sensing imagery was suggested by Welch and Salter [1]. However,

their study utilizes simulated digital data produced by manual interpretation of aerial photography. During the past year, Fu and Yu have applied the Welch and Salter approach to multispectral scanner data collected by aircraft and spacecraft. The results of these experiments confirmed that contextual information can be extracted from multispectral imagery by this approach and that such information aids in achieving accurate classification.

But the Welch and Salter method suffers from a number of practical difficulties. The first is the severe restrictions placed on the form of the spatial context that can validly be incorporated because of the simplifying assumptions which are necessary. Essentially, only the data associated with the four "nearest neighbors"

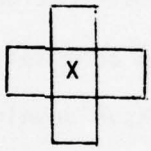


Fig. 1. Image point to be classified (X) and its four nearest neighbors.

of a point to be classified (Figure 1) can be accounted for. The second difficulty is the added computational load required to implement the method and apply it to every image point.

Two new models which we have formulated for the two-dimensional random process, based on a significantly different way of thinking about the spatial dependencies, hold promise for alleviating both of these problems. Arbitrary contextual configurations, such as those illustrated in Fig. 2, can be dealt with by means of the new models, and the required computations are somewhat simpler. Furthermore, the statistical dependencies which must be learned from a typical scene in order to classify a new scene are less complex

and more readily determined.

To further reduce the computational load without compromising the quality of the analysis results, we are considering the use of sequential and adaptive processes which will determine regions of an image for which context analysis is potentially useful. When there is no potential benefit to be gained from

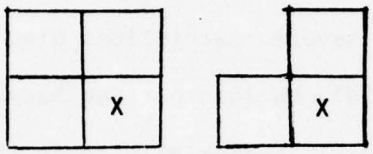


Figure 2. Alternative contextual configurations

incorporating context into the classification process, simpler decision processes can be applied, reducing the computations accordingly.

During this project year, we have developed software which will enable us to statistically characterize the contextual content of a LANDSAT image. In addition, we have formulated and begun implementation of a Bayesian classifier model which is particularly attractive for contextual classification. This classifier is also useful for analyzing time-sequential image data and its use for that purpose is soon to be demonstrated.

REFERENCE

[1] J. R. Welch and K. G. Salter, "A context algorithm for pattern recognition and image interpretation," IEEE Trans. on Systems, Man and Cybernetics, vol. SMC-1, No. 1, January 1971.

PHASE UNWRAPPING

B. O'Connor and T.S. Huang

1. Introduction

In several applications of one and two-dimensional signal processing, the analytic phase function must be computed. This function is called the continuous or unwrapped phase. One well known use of the unwrapped phase is to compute the complex cepstrum of a signal. Cepstral techniques have been used on one-dimensional signals in such areas as seismic, speech, and hydrophonic analysis [1]. More recently, these techniques have been extened to two-dimensions with applications to stability [2,3].

Other non-cepstral applications of the unwrapped phase include blind deconvolution and polynomial root distribution determination. Blind deconvolution refers to methods which estimate a blurring function given only the blurred picture. Successful work has been reported for [4,5,6,7] estimating the magnitude of the blurring function from detailed blurred images. However, the determination of the phase has not been adequately solved yet. Pattern recognition techniques have been applied with some success when the blurring function is one from a class of several [8]. A more general method, which is a modification of a technique applied to multi-frame processing, has been a disappointment judging from preliminary results [10,11]. Phase unwrapping has been used to reconstruct the phase of the one-dimensional blurring function from a blurred image [7], again with the disappointing results. However, the results were obtained from an unsatisfactory phase unwrapping algorithm. The employed algorithm failed in many cases to unwrap the phase correctly. Hence, the conclusions reached were premature. Recently, a new phase unwrapping algorithm has been proposed [12]. This method combines the

information in both the phase derivative and the principal value of the phase into an adaptive numerical integration scheme. This algorithm has the potential of being very accurate. However, it did not obtain the reliability claimed in the paper. Below we will discuss among other things our improvements to this algorithm. We have found a way to increase both the speed and the accuracy of this phase unwrapping method. Our modified algorithm can now be applied to the phase reconstruction problem to obtain accurate results. We plan in the future to attack the blind deconvolution problem in both one and two dimensions and hopefully more reliable conclusions will result. A previous report [3] discussed the two-dimensional phase unwrapping problem.

II. Phase Unwrapping (One Dimension)

Let $x(n)$ be a finite real sequence of numbers $n \geq 0$ with Fourier transform $X(e^{j\omega})$, the unwrapped phase of X is $\arg[X(e^{j\omega})]$. This must be a continuous and odd function of ω. Generally, only $ARG[X(e^{j\omega})]$ (the principal value) can be evaluated from $X(e^{j\omega})$ via the inverse tangent. That is, only the principal value of the phase is known and this must be unwrapped to obtain $\arg[X(e^{j\omega})]$. If $X(e^{j\omega})$ were known for all ω, this would be a trivial task. All that would be necessary is to patch together pieces of $ARG[X(e^{j\omega})]$ to obtain a continuous function with boundary condition $\arg[X(e^{j0})] = 0$. However, the FFT (Fast Fourier Transform) is employed in the calculation of $X(e^{j\omega})$ and hence, $X(e^{j\omega})$ is known only on a discrete set of points. Thus, the discontinuities in $ARG[X(e^{j\omega})]$ are difficult to detect. One method of phase unwrapping examines the samples of the principal value of the phase and attempts to detect large jumps between adjacent samples. The rationale is that these jumps depict the discontinuities of $ARG[X(e^{j\omega})]$. However, practical problems, such as what can be considered a large jump and a proper sampling rate, plague this technique. Generally, for this method to have any

chance of success at all, it is necessary to sample the phase at a very high rate. However, a high sampling rate does not guarantee success and it results in excessive calculation time. As an example, Filip [7] used a 2048 length FFT to unwrap the phase of a sequence of length 128 with the jump threshold set at 1.06 pi. We tried his algorithm on 100 sequences of length 128 obtained from a picture of various texture patterns and found that it failed 29% of the time.

The unwrapped phase can be unambiguously defined in terms of its derivative $\arg'[X(e^{j\omega})] = \frac{d}{d\omega} \arg[X(e^{j\omega})]$ by $\arg[X(e^{j\omega})] = \int_0^\omega \arg'[X(e^{js})]\, ds$, $\arg[X(e^{j0})] = 0$. Computation of the phase derivative is straightforward [1,2]

$$\arg'[X(e^{j\omega})] = \frac{X_R(e^{j\omega})X_I'(e^{j\omega}) - X_I(e^{j\omega})X_R'(e^{j\omega})}{|X(e^{j\omega})|^2}$$

Note that $X'(e^{j\omega}) = -jFT\{nx(n)\}$. Hence, the integration of the phase derivative is another technique for phase unwrapping. The advantage of this method over the previous one is that the detection of discontinuities is no longer necessary. However, this method suffers from problems due to numerical integration. Again, high sampling rates are necessary to produce acceptable results. Furthermore, the phase error increases as ω increases as a result of accumulating integration errors. Higher order integration formulas do not offer a solution to this problem because they introduce new problems [13].

A logical alternative is to combine the information in the principal value with that of the phase derivative. Tribolet [12] successfully accomplished this by employing an adaptive numerical integration scheme. He used trapezoidal integration to acquire a number which is added to the previous value of the unwrapped phase. The principal value of this estimate is compared to the known principal value. If these numbers are close enough, the

estimate is clamped to the appropriate value of the unwrapped phase which has the proper principal value. In other words [12], assuming the unwrapped phase at ω_0 is known, we define an estimate of the unwrapped phase at $\omega_1 > \omega_0$ by:

$$\tilde{\arg}[X(e^{j\omega_1})|\omega_0] = \arg[X(e^{j\omega_0})] + \frac{\omega_1-\omega_0}{2}\{\arg'[X(e^{j\omega_0})] + \arg'[X(e^{j\omega_1})]\}$$

Clearly this estimate improves as the step interval $\Delta\omega = \omega_1-\omega_0$ becomes smaller. The basic idea of Tribolet's algorithm is to adapt the step size $\Delta\omega$ until the result of numerical integration matches the information given by the principal value of the phase. Define $E(\omega_0,\omega_1) \triangleq \tilde{ARG}[X(e^{j\omega_1})|\omega_0] - ARG[X(e^{j\omega_1})]$. The step size which leads to a consistent phase estimate at ω_1 is one in which $|E(\omega_0,\omega_1)| < \overline{E} << \pi$. When the above condition is satisfied the unwrapped phase is defined by:

$$\arg[X(e^{j\omega_1})] = \tilde{\arg}[X(e^{j\omega_1})|\omega_0] - E[\omega_0,\omega_1]$$

so it wraps to $ARG[X(e^{j\omega_1})]$ without error. Tribolet also wrote a clever program which implements this algorithm. He takes full advantage of the FFT algorithm and reduces the number of extra DFT computations to a reasonable small number. The FFT algorithm is used to calculate $X(e^{j\omega})$, $X'(e^{j\omega})$, $ARG[X(e^{j\omega})]$ and $\arg'[X(e^{j\omega})]$ at $N = 2^M$ equally spaced frequency points for N greater than or equal to the total number of points in x(n) input sequence. The above scheme is used to unwrap the phase starting at $\omega = 0$. The step size adaption was designed to minimize the number of extra DFT's required. As will be shown later this is necessary in order to keep the computation time reasonable.

III. Modification of Tribolet's Phase Unwrapping Algorithm

A simple modification of Tribolet's algorithm will allow the phase to be unwrapped with greater reliability and speed. One of the major difficulties with phase unwrapping algorithms occur when the input sequence has zeroes close to the unit circle. Near these points [see appendix A] the phase and the phase derivative change rapidly thus causing errors in the standard methods of phase unwrapping. Tribolet's algorithm effectively combats these errors in many cases by adaptively integrating near such points. However, large unwrapping errors can occur when the integrated phase derivative clamps to an incorrect value of the unwrapped phase. These errors can be spectacular at times causing the unwrapped phase to jump to values off by several thousand. In its present form Tribolet's algorithm produces these errors quite often when there exist zeroes very close to the unit circle. Furthermore, these errors are essentially independent of the size of the FFT. Ideally, a phase unwrapping algorithm should either unwrap the phase to the correct value or fail. Failure should occur when zeroes are too close to the unit circle. How close, should be an option determined by the user's application and the computer system's accuracy.

This reliability problem can be traced to the calculation of the phase derivative's integral between two adjacent points. If the phase derivative is changing rapidly in this region, the integral's value may be quite large. Furthermore, this value may be such that when added to the previous value of the unwrapped phase to give an approximation of the new value of the unwrapped phase, its principal value may be close enough to the true principal value (determined by the parameter $\overline{E}$) for it to be clamped to an incorrect value. This can occur before the algorithm has a chance to either initiate adaptive integration or to carry adaptive integration further.

One possible solution is to decrease the clamping threshold value $\overline{E}$. A smaller value of $\overline{E}$ will reduce the probability of this kind of error because a smaller clamping region exists around the principal value. However, too small a magnitude will cause many unnecessary adaptions and, hence, increase the computational time excessively. (Note, $\overline{E} = \pi/4$ in Tribolet's program.) A better solution is to limit the integrai's magnitude to jumps less that π. If a jump is larger than this then start adaption by temporarily halving the step size. This will totally eliminate false jumps and it will allow adaptive integration to do its job and find the true unwrapped phase. With this method $\overline{E}$ can be set to a larger value such as $\pi/4$ so that the number of unnecessary adaptions will be minimized. Note, in some cases this will add unnecessary computation when the phase difference $\omega_k - \omega_{k-1}$ has magnitude greater than π and at the same time, the integrated phase derivative indicates this value of ω_k. The occurrence of this is extremely rare. This will be demonstrated later when the results on Table 1 are discussed.

An important consideration is the choice of FFT size N, that is, the number of samples of $X(e^{j\omega})$, $X'(e^{j\omega})$, and $ARG[X(e^{j\omega})]$. It is generally true that the longer the input sequence x(n) the higher the probability that it has zeroes close to the unit circle. Hence, since the complex cepstrum falls off as α_k^n/n or β_k^{-n}/n where α_k is the magnitude of the root which is both less than one but closest to one, and β_k is the magnitude of the root which is both greater than and closest to one, the complex cepstrum will have a longer duration. Moreover, since the Fourier transform of the odd part of the complex cepstrum is proportional to the unwrapped phase, therefore, the unwrapped phase can be more oscillatory than the magnitude of the Fourier transform. Hence, by the inverse sampling theorem, it is desirable to unwrap the phase on a grid of points more numerous than the number of input sequence

points to avoid gross undersampling errors. Note that the complex cepstrum usually has infinite duration so theoretically the unwrapped phase cannot be reconstructed exactly from its samples. However, since the cepstrum decays rather quickly an adequate sampling interval can be found.

Since the algorithm uses adaptive integration, it would seem that the sampling rate is not important if the only desire is to unwrap the phase at these specific samples. However, the number of samples is important for three reasons. First, usually the samples of the unwrapped phase are calculated to represent the continuous unwrapped phase. Hence, from the above discussion, the number of frequency samples must be at least several times more than the length of input sequence. Secondly, if the sampling rate is low, then the number of adaptions and hence extra DFT calculations will be high. As demonstrated in appendix B, if the number of adaptions is high, then it will be more efficient to increase the sampling rate by using a larger FFT size. Finally, a low sampling rate can cause unwrapping errors, even when the modified adaptive algorithm is employed. For example, two adjacent frequencies may have the same principal values and equal and opposite phase derivatives. Hence, the algorithm will unwrap the phase of the second to that of the first. However, it is possible that its actual value is plus or minus an integral multiple of two pi. In this example, the adaption routine will never have a chance to correct this undersampled condition. This example is indeed a limiting case for any phase unwrapping technique in which the phase is undersampled. However, the probability of occurrence of such errors can be lowered by either increasing the FFT size or by lowering the clamping threshold $\overline{E}$. Unfortunately, lowering $\overline{E}$ increases the number of adaptions greatly and, hence, raises computation time excessively.

Some experimental results are indicated in table I which demonstrates the tradeoff of FFT size and number of adaptions. One hundred sequences of length 256 were generated from random numbers of magnitude less than one. For most of the algorithms tested FFT sizes were varied from 256 to 2048 in multiples of two. For each FFT size the below information was recorded. The first entry is the number of sequences in which the algorithms failed to unwrap the phase either correctly or incorrectly. Failures can result from either a zero being located on the unit circle or if a frequency interval has been divided into too many parts during adaption. For our program we allow an interval to be subdivided into 65536 parts. The next entry indicates the number of phases (out of one hundred) which were unwrapped incorrectly. Methods for obtaining this information will be discussed shortly. The following number represents the average number of adaptions. The last entries give the average number of multiplications necessary to unwrap the phase. This should give a rough estimate for how long a FORTRAN program would take to unwrap the phase of a sequence in which the number of adaptions is equal to the indicated average. However, since such operations as FFT resorting and the inefficient machine code generation of FORTRAN are not included in this quantity, it does not portray an accurate indication of a FORTRAN program's speed. Nevertheless, the number of multiplications would give an accurate estimate of speed if the algorithm was implemented in digital hardware. A more complete discussion on the algorithm's efficiency is presented in appendix B.

We first tested Tribolet's algorithm for three different values of $\overline{E}$ and several different FFT sizes. Notice errors still exist when $\overline{E} = \pi/16$ and the FFT size is 2048. In many cases the accumulated error caused the phase to be off by as much as twenty pi. Also observe that the average number of

adaptions increased as $\overline{E}$ is decreased (approximately a two fold increase for each time $\overline{E}$ is halved) while the number of correct phase unwrappings increased. As $\overline{E}$ is decreased the percent of failure increased because the probability that the integrated phase enters the clamping region is lowered.

Now when Tribolet's algorithm is modified to include the integrated phase jump limit of pi the accuracy improved greatly. For FFT sizes greater than 256, no errors occurred for these examples. A few errors did occur when the FFT size was 256 due to extreme undersampling. A close study of the table reveals that the number of adaptions is approximately halved for each two fold increase in FFT size. Since the number of calculations for a FFT is proportional to $N \cdot \log_2 N$ and since each adaption requires a multiple of N further operations, there exists an optimal FFT size which minimizes the total number of operations for a specific example. However, since knowledge about the number of necessary adaptions is not available before hand, only a rule of thumb can be formulated. For the data on table I, FFT sizes of four times the sequence length gave the fastest computer runs. However, this is data dependent and in fact if we assume all the input sequence entries are positive such as picture data (see Table II) FFT sizes eight times larger gave the fastest results.

We have changed Tribolet's algorithm even further to employ a modified Simpson's integration rule which results in a 10-20% reduction in the average number of adaptions without noticeable affecting the algorithm's accuracy (see Table I). One explanation of this result is that in many cases one application of Simpson's rule is better than two applications of trapezoidal integration [13] and hence further adaptions can occasionally be avoided. Higher order integration schemes cannot be applied to this problem.

A further study of tables I and II will reveal some other interesting results. For picture data if the jump limit is decreased to $\pi/2$ for FFT size of 512 approximately half the number of errors occurred. However, the number of adaptions increased. Note that this example was presented for only instructive purposes since FFT sizes of 1024 and 2048 give much fewer operations. If the jump limit is increased to 5.5 for the random data, errors appeared when FFT size was 512. This can be explained by noting that the higher the jump limit is, the higher the probability of a false jump.

IV. Error Analysis

In order to evaluate the accuracy of the discussed phase unwrapping algorithms a method must be found which is capable of unwrapping the phase with no error. However, no practical analytic technique exists for accurately determining the unwrapped phase for a long sequence at all sample frequency points. Fortunately, the unwrapped phase at $\omega = \pi$ can be calculated from any of a number of techniques. The unwrapped phase at $\omega = \pi$ can be compared to this true value to determine if any errors occurred in the phase unwrapping procedure. Hence, false jumps can be detected. The reader should note that the equality of these numbers does not ensure that the phase was unwrapped correctly, because errors can be both positive and negative, so offsetting errors can exist. Fortunately, the probability of offsetting errors is small. Notice, further, that if the unwrapped phase at $\omega = \pi$ does not equal the true value then phase unwrapping errors definitely exist.

Methods which can calculate the unwrapped phase at $\omega = \pi$ are derived from the fact that $\arg[X(e^{j\omega})] = -m_0\pi$. Here m_0 is the number of zeroes outside the unit circle which the polynomial formed from the input sequence has. Hence, one obvious method to determine m_0, is to find all the roots of this

polynomial. However, the input sequences are usually quite long so root finding techniques become too time consuming. Of all the studied techniques we found that a modified Jury table [14] was the easiest to implement and had fewer operations than most. The Jury table method does have numerical problems for large polynomials so it was necessary to use double precision arithmetic. See table I for a tabulation of results. In summary, the Jury table was employed to calculate m_0 which, in turn, is used to detect errors that occur in the phase unwrapping process.

For long sequences still another method exists which can lower the number of adaptions on the average of approximately 10 percent. Generally, an aribitray sequence of numbers has a phase which contains a significant linear component. If this component could be reduced or eliminated, then this should decrease the number of adaptions. This is especially true when the phase unwrapping algorithm has a phase jump limit. Here, the linear phase component's slope contributes to the integration phase derivative and, hence, increases the probability that its value will surpass the phase jump limit. Therefore, the linear phase component can introduce unnecessary adaptions which would not occur if this component was smaller. Unfortunately, for a given sequence the linear phase component is not known. However, for a sequence of length L the average slope of this line is approximately L/2. This follows from the fact that this slope is equal to the number of zeroes outside the unit circle. Furthermore, thousands of experiments on picture and random data corroborate this. Therefore, if all input sequences are shifted by L/2 before phase unwrapping, this component will be reduced and the number of adaptions will be reduced. See table I for an example of this.

The importance of reducing the number of adaptions must be emphasized. Each adaption requires the calculation of a DFT at a particular frequency.

A large number of extra DFT calculations results in excessive computational time. Hence, methods which reduce this number are important. Above, we have discussed methods which can both increase the accuracy and decrease the computational time for adaptive phase unwrapping algorithms.

Work is presently being performed to use the above techniques to characterize the phase of textures. Hopefully, several parameters can be extracted from the unwrapped phase which can adequately describe the texture.

REFERENCES

[1] A. Oppenheim and R. Schafer, Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.

[2] M. Ekstrom and J. Woods, "Two-Dimensional Spectral Factorization with Application in Recursive Digital Filtering," IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. 24, pp. 115-127, April 1976.

[3] B. O'Connor and T. Huang, "Two-Dimensional Complex Cepstrum," Image Understanding and Information Extraction, Tech. Rep., pp. 103-108, May-July 1976.

[4] T. Stockham, "Blind Deconvolution Through Digital Signal Processing," Proc. IEEE, pp. 678-692, April 1975.

[5] E. Cole, "The Removal of Unknown Image Blurs by Homomorphic Filtering," Computer Science, University of Utah, UTEC-CSC-74-029, June 1973.

[6] T. Cannon, "Digital Image Deblurring by Nonlinear Homomorphic Filtering," Computer Science, University of Utah, UTEC-CSC-74-091, August 1974.

[7] A. Filip, "Estimating the Impulse Response of a Linear, Shift-Invariant, Image Degrading System," Ph.D. Thesis, M.I.T., October 1972.

[8] M. Cannon, "Blind Deconvolution of Spatially Invariant Blurs with Phase," IEEE Trans. Acoustics, Speech, and Signal Processing, Vol. ASSP-24, pp. 58-63, February 1976.

[9] K. Knox and B. Thompson, "Recovery of Images from Atmospherically Degraded Short-Exposure Photographs," Astrophysical Journal, vol. 193, No. 1, Oct-Nov. 1974.

[10] B. O'Connor and T. Huang, "Estimating the Impulse Response of a Degrading System," Image Analysis and Modeling, Purdue University, RADC-TR-76-76, pp. 51-52, March 1976, (A023633).

[11] J. Morton, "A Technique of A Posteriori Restoration,", USC-Semiannual Technical Report, pp. 121-130, Sept. 1976.

[12] J. Tribolet, "A New Phase Unwrapping Algorithm," to appear in IEEE Trans. on ASSP.

[13] B. Carnahan, H. Luther, and J. Wilkes, Applied Numerical Methods, John Wiley and Sons, Inc., N.Y., 1969.

[14] G. Maria and M. Fahmy, "On the Stability of Two-Dimensional Digital Filters," IEEE Trans. ASSP, pp. 470-472, October 1973.

APPENDIX A

Effect On Phase Unwrapping When Zeroes Are Near The Unit Circle

Let $X(z)$ be the Z-transform of a finite length sequence $x(n)$. Then

$$X(z) = Az^{r} \prod_{k=1}^{M} (1-a_k z^{-1}) \prod_{k=1}^{N} (1-b_k z)$$

where all $|a_k| < 1$ and $|b_k| < 1$, here we are not allowing zeroes on the unit circle. Hence, the Fourier transform is defined as follows:

$$X(e^{j\omega}) = A\, e^{j\omega r} \prod_{k=1}^{M} (1-a_k e^{-j\omega}) \prod_{k=1}^{N} (1-b_k e^{j\omega})$$

For simplicity assume all zeroes are within the unit circle so

$$X(e^{j}) = \prod_{k=1}^{M} (1-a_k e^{-j\omega})$$

which implies

$$|X(e^{j\omega})| = \prod_{k=1}^{M} |1-a_k e^{-j\omega}|$$

$$\arg[X(e^{j\omega})] = \sum_{k=1}^{M} \mathrm{ARG}(1-a_k e^{-j\omega})$$

Let

$$a_k = |a_k|\, e^{j\omega\theta_k}$$

then

$$\arg[X(e^{j\omega})] = \sum_{k=1}^{M} \mathrm{Arctan} \left\{ \frac{|a_k| \sin(\omega-\theta_k)}{1-|a_k| \cos(\omega-\theta_k)} \right\}$$

Consider only the ℓth term which corresponds to a zero close to the unit circle. Assume $|a_\ell| = 1-\varepsilon$ where ε is positive and small. Now

$$\frac{d}{d\omega}\left[\text{Arctan}\ \frac{|a_\ell|\sin(\omega-\theta_\ell)}{1-|a_\ell|\cos(\omega-\theta_\ell)}\right] = \frac{|a_\ell|(\cos(\omega-\theta_\ell)-|a_\ell|)}{1-2|a_\ell|\cos(\omega-\theta_\ell)+|a_\ell|^2}$$

Hence, it follows that at $\omega = \theta_\ell$ the phase due to this zero is zero and the phase derivative is a maximum approximately equal to ε^{-1}. At $\delta = \sqrt{2\varepsilon}$ on either side of $\omega = \theta_\ell$ the phase derivative is zero and the phases are either approximately $\pm \pi/2$. Therefore, in the interval $\theta_\ell - \sqrt{2\varepsilon}$ to $\theta_\ell + \sqrt{2\varepsilon}$ the phase increases by approximately π while the phase derivative starts at zeroes increases rapidly to $1/\varepsilon$ and then decreases to zero again. In fact, in the region the phase derivative has an average value equal to $\pi/(2\sqrt{2\varepsilon})$. Of course, in order to obtain the true phase and phase derivative of $x(n)$ the contributions of all the zeroes and poles must be added.

Hopefully, the above gives some incite into some possible problems which plague phase unwrapping methods. When a sequence has zeroes close to the unit circle both the phase and phase derivative change rapidly near these frequencies. Hence, a higher sampling rate is required near such points to adequately represent these quantities. Adaptive phase unwrapping has the potential of sampling the phase and phase derivative more often around these frequencies while sampling less when the phase is slowly varying. However, Tribolet forgot to protect his algorithm from the integrated phase clamping itself to those values of phase which constitue an excessive jump.

APPENDIX B

Computational Considerations For Adaptive Phase Unwrapping

Below, we plan to calculate the number of multiplications necessary to implement adaptive phase unwrapping and to discuss computational tricks which can reduce this number. Both trapezoidal and Simpson's integration schemes will be considered. Basically, adaptive phase unwrapping can be broken into two parts. The first consists of using the FFT and arc tangent to determine both ARC[$X(e^{j\omega})$] and arg'[$X(e^{j\omega})$] on a uniform circular grid around the unit circle which, in turn, is used to calculate arg[$X(e^{j\omega})$] on this same grid. The second part of phase unwrapping is necessary if part one fails to meet the consistency condition. Here, the DFT is employed to calculate the principal value of the phase and phase derivative at points not on the initial grid until a consistent value is found. This is the adaptive part of phase unwrapping and its deployment is data dependent.

During the initial stages of phase unwrapping both the principal value and phase derivative must be calculated. Assume $x(n)$ is a finite sequence of length M and let $N \geq M$ (N is a power of two) be the number of points computed in the FFT. In order to determine the above quantities both $X(e^{j\omega})$ and $X'(e^{j\omega})$ need to be computed. $X(e^{j\omega})$ is the Fourier transform of $x(n)$ while $X'(e^{j\omega})$ is the Fourier transform of $-j\,n\,x(n)$. Both X and X' can be found using only one application of the FFT since $x(n)$ and $n\,x(n)$ are real sequences. Just let $s(n) = x(n) + j\,nx(n)$ then $X(e^{j\omega}) = 1/2(S(e^{j\omega}) + S^*(e^{-j\omega}))$ and $X'(e^{j\omega}) = -1/2(Se^{j\omega}) - S^*(e^{-j\omega}))$. This trick allows two Fourier transform calculations in one FFT and hence only $2N\log_2 N$ real multiplications are required plus 2M multiplications to form $-nx(n)$. Since the unwrapped phase is an odd function, it need only be unwrapped at N/2 points. At each of these

points the following calculations must be performed: the phase derivative which requires 4 multiplications; the principal value of phase which requires approximately 8 multiplications; trapezoidal integration and a consistency check which need approximately 4 multiplications. Hence, the modified Tribolet's algorithm requires $2M + 2N \log_2 N + 8N$ real multiplications for its nonadaptive part. Our algorithm which employs Simpson's integration requires approximately $2M + 2N \log_2 N + 10N$ multiplications for this stage.

The number of operations necessary for the second stage, the adaptive part, of phase unwrapping is data dependent. It depends on A the average number of adaptions necessary to unwrap the phase of a set of sequences. This number is also dependent on the FFT size N and, in fact, for every two-fold increase in N A is approximately halved. For each adaption the following operations are necessary: 14 multiplications to calculate both a sine and cosine, 8 for the arc tangent, 5 for the phase derivative, $7 \cdot M$ for DFT, and approximately 13 other operations. Therefore, the number of real multiplications for adaption is approximately $7 \cdot A \cdot (M+6)$. Hence, the number including parts one and two totals $2M + 2N \log_2 N + 10N + 7 \cdot A \cdot (M+6)$. It should be stressed that his isn't necessarily an accurate guide for determining the speed of a FORTRAN program. Other factors such as assembly code optimization and floating point hardware enter into this calculation. However, this formula does indicate that there exists an optimum FFT size N which can minimize operations since A is inversely proportional to N. Also, it is advantageous to find a method to minimize A for a given N. In terms of computer time, we found that for the random number data $N = 4M$ was best while for picture data $N = 8M$ gave comparably accurate results. When the total number of calculations is compared to computer time for a FORTRAN program we found that in order to correlate closely the nonadaptive calculations must be weighted heavier.

The above implementation is more efficient and accurate than Tribolet's [12] original adaptive phase unwrapping algorithm since Simpson's integration rule and a phase jump limit were employed. Another simple programming change was implemented which increased the speed significantly, by pulling out a complex exponential calculation out of DFT loop in the adaptive stage of the program. This obviated its repeated calculation but introduced numerical problems which were overcome by the use of double precision arithmetic.

APPENDIX C

Phase Unwrapping From Principal Value Only (Filip's Method)[7]

In this appendix phase unwrapping from only principal value information is compared to adaptive phase unwrapping. As stated in the main report, in order for the phase to be unwrapped from principal value samples it must be sampled at a very high rate. Furthermore, a threshold must be chosen which determines the existence of principal value discontinuities. Hence, both the sampling rate and the threshold are important parameters for this method of phase unwrappings. Unlike adaptive phase unwrapping this technique does not use either intersample or phase derivative information, and as a result it is not as accurate. This is especially true when the input sequence has zeroes near the unit circle. Furthermore, since adaptive phase unwrapping employs FFT sizes only several times (usually 4 to 8) the original sequence length, while the above method needs ratios of 16 to 32 times, hence, adaptive phase unwrapping is faster as well as more accurate.

See table III for tabulated results of phase unwrapping from principal value information only. Each line summarizes the results of unwrapping the phase of one hundred sequences of length 128. For the first half of the table, the sequences were obtained from a picture which contained many random texture patterns. The numbers ranged from 0 to 255. These examples were used to test the accuracy of Filip's phase unwrapping results on picture data [7]. He used a 2048 length FFT to unwrap the phase of a sequence of length 128 with the jump threshold set at 1.06π. For these parameters we found that 30 percent of the sequences had their phase unwrapped incorrectly. That is, either the algorithm indicated a false jump (represented by negative numbers) of 2π or it found a false discontinuity causing a positive error of 2π. Note that

several of these errors can occur in the phase unwrapping of one sequence and, hence, there is a small probability that offsetting errors may occur. For the second half of the table, the sequences were formed from a random number generator which calculated numbers between minus one and one.

Filip's phase unwrapping algorithm is plagued by some important practical problems. Most phase functions contain a linear component of slope equal to the number of zeroes outside the unit circle if the input sequence is considered to be a polynomial in z^{-1}. The effect of this component is to increase the probability of missing a discontinuity for a given threshold.since the phase will change more between points. (See no shift cases in table.) There are several ways of surmounting this problem. First we could try to converge to the correct phase by iteratively removing the linear phase component. This method was used by Filip [7]. He initially unwrapped the phases of the given sequences and noted the value at $\omega = \pi$. This value divided by π gave an approximation to the slope of the linear phase component. Next, he effectively subtracted this component by shifting the input picture segment by this amount. Finally, these steps were repeated several times until the unwrapping algorithm had value of zero for the phase at $\omega = \pi$. The speed of this algorithm could be increased if all the initial sequences were shifted by half the sequence length. This is motivated by the fact that the number of zeroes outside the unit circle for a large number of sequences averages around this value. Hence, fewer time consuming iterations will be needed. For sequences of length not more than 256 it would be much quicker to use a method such as the Jury table to determine the zero distribution of the sequence and hence the true linear phase slope. This obviates the need to calculate the phase more than once. Hence, the entries in the table which indicate a shift were calculated by first using the Jury table to determine the linear phase

slope and then each sequence was shifted by the appropriate amount to remove this component. However, even if this is known accurately Filip's phase unwrapping algorithm still is error prone as seen in the table. Even for a FFT size of 4096 errors occur. On the other hand, FFT sizes of 512 are quite adequate for obtaining accurate results for adaptive phase unwrapping. Furthermore, the linear phase component need not be found since the phase derivative takes this into account.

Filip's phase unwrapping algorithm employs approximately

$$2N \log_2 N + \frac{N}{2} (8+2)$$

multiplications (8 multiplications for ATAN2) for each iteration. The Jury table needs only M^2 multiplications, where M is the sequence size. For picture data N = 32M gives about 95% accuracy for the examples used. Hence, for a sequence of length 128 approximately 135,000 multiplications are needed to unwrap its phase. For such sequences adaptive phase unwrapping requires

$$2M + 2N \log_2 N + 10N + 7 \cdot A \cdot (M+6)$$

(see appendix B) multiplications. For picture data A is approximately 36 while for random number data A is approximately 8.5 so 65,000 and 42,000 multiplications are needed.

TABLE I

Random Number Data (100 256 length sequences)

Algorithm	FFT Size	% Failed	% Errors	A	No. of Calculations (1000's) Stage 2	Stage 1	Total
Tribolet	256	3	69	144	265	7	272
$\bar{E} = \pi/4$	512	3	43	81	149	13	162
No jump limit	1024	3	26	42	77	28	105
	2048	3	15	21	39	63	102
Tribolet	512	5	32	139	255	13	268
$\bar{E} = \pi/8$	1024	5	18	73	134	28	162
No jump limit	2048	5	9	36	66	63	129
Tribolet	512	10	17	221	405	13	418
$\bar{E} = \pi/16$	1024	10	11	121	222	28	250
No jump limit	2048	10	5	62	114	63	177
Modified	256	3	9	186	340	7	347
Trapezoidal	512	3	0	90	165	13	178
$\bar{E} = \pi/4$	1024	3	0	47	86	28	114
jump limit = π	2048	3	0	21	39	63	102
Simpson	256	0	50	175	320	7	327
Method	512	0	0	79	145	14	159
$\bar{E} = \pi/4$	1024	0	0	36	66	31	97
jump limit = π	2048	0	0	17	31	66	97
Shift of 128	1024	0	0	33	60	31	91
Simpson	256	2	67	138	253	7	260
$\bar{E} = .7$	512	2	12	70	128	14	142
jump limit = 5.5	1024	2	0	35	64	31	95
	2048	2	0	18	33	66	99
Jury table (double precision)		0	0		15% longer than single (PDP-11-45)		
Jury table (single precision)		0	5				131

TABLE II

Picture Data (100 256 length sequences)

	FFT Size	% Failed	% Errors	A	No. of Calculations (1000's) Stage 2	Stage 1	Total
Simpson	512	1	24	342	627	14	641
$\overline{E} = \pi/4$	1024	1	0	139	255	31	286
jump limit = π	2048	1	0	65	119	66	185
Simpson							
$\overline{E} = \pi/4$	512	0	11	450+	800+	14	800+
jump limit = $\pi/2$							
Modified	512	1	0	317	581	13	594
Trapezoidal	1024	1	0	163	299	28	327
$\overline{E} = \pi/4$	2048	1	0	85	156	63	219
jump limit = π							
Jury table (double precision)		2					131 double precision

TABLE III

Phase Unwrapping
only using principal value (Filip's method)

	FFT Size	Threshold	% in Error	Sum of positive and negative errors	Sum of absolute value errors
Texture Data					
with shift	2048	1.06 π	30	-28	84
with shift	2048	π	16	-10	34
with shift	2048	.9 π	49	- 4	132
with shift	4096	π	5	- 2	10
with shift	1024	π	42	- 6	104
no shift	2048	π	66	-216	216
Random Number Data					
with shift	2048	π	1	2	2
with shift	1024	π	3	2	6
with shift	512	π	18	4	36
no shift	2048	π	48	-108	108

THE PROJECTION METHOD

S. P. Berger and T. S. Huang

I. Introduction

The projection algorithm is an iterative approach to image restoration. The method has been extensively described in previous reports, but a brief explanation will also be presented here. Two studies have been completed to compare the projection method with the singular value decomposition approach. A two-dimensional image was used in one study which has been reported earlier. More recent results have used a one-dimensional signal to compare the two methods. This report included a discussion of plans for the application of the projection method to mensuration and classification of actual images. Other anticipated results are also discussed.

II. The Algorithm

The purpose of a restoration process is to approximate the original image as closely as possible. The degradation of a two-dimensional image can be represented by $g(x,y) = D[f(x,y)] + n(x,y)$, where $f(x,y)$ is the original image, $n(x,y)$ is noise, D is a degrading operator, and $g(x,y)$ is the degraded image. The discretized version of an image consists of a finite array of picture elements (pixels) with magnitudes representing gray levels.

If the degradation is linear, it can be represented as $g = [H]\, f + n$, where we have "stacked" the rows of the image array into a column vector f. Neglecting noise the equations are

$$g_1 = h_{11}f_1 + h_{12}f_2 + \cdots + h_{1N}f_N$$

$$g_2 = h_{21}f_1 + \cdots \quad + \quad + h_{2N}f_N$$

$$\vdots$$

$$g_M = h_{M1}f_1 + \cdots \quad + \quad + h_{MN}f_N$$

where N may not equal M. The projection algorithm is an iterative technique for solving this system of equations.

The solution is obtained by successive projection onto planes in hyperspace. For the case where a unique solution does exist, the algorithm will converge to the point of intersection of the hyperplanes. If the planes do not intersect at a single point, the algorithm will converge to a point which may be a useful approximation in a restoration sense. One cycle of iterations is complete when the projection is made onto the Mth hyperplane.

Since the algorithm was compared with the singular value decomposition (SVD) approach, a review of the SVD will be given here. The SVD method is based on a representation of the pseudo-inverse of a matrix. The linear degradation has the form $g = [H]\, f + n$ for the discrete case. The original vector f can be estimated by $\hat{f} = [H]^{+}g$, where $[H]^{+}$ is the Moore-Penrose pseudo-inverse. Now $[H]^{+}$ can be obtained by

$$[H]^{+} = \sum_{i=1}^{R} \frac{1}{(\lambda_i)^{1/2}} V_i U_i^{T}$$

where R is the rank of [H], U_i and V_i contain (in columns) the eigenvectors of $[H][H]^{T}$ and $[H]^{T}[H]$, respectively. The restoration is then given by

$$\hat{f} = \sum_{i=1}^{P} \frac{1}{(\lambda_i)^{1/2}} V_i U_i^T g \ .$$

The user must decide on the optimum value of P, since the effect of noise will dominate the summation after a certain number of terms.

III. Past Results

A comparison between the projection algorithm and the SVD which was reported earlier involved an 8 x 8 element representation of the number "5". A linear degradation with additive Gaussian noise was simulated. The value at each pixel was replaced by the average of the 9 points in the 3 x 3 block surrounding the element. The SVD approach to this problem was given by Huang and Narendra [1].

The results showed that the projection method performed as well or better than the SVD for low additive noise levels. However, for high noise levels, the subjective appearance of the SVD restored images was slightly better than the projection results.

The main difficulty with the SVD is the generation and storage of the eigenvectors and eigenvalues. Even for the small 8 x 8 image, the eigenvector matrix is 64 x 64. For large images the difficulties will be more pronounced. The storage requirements for the projection algorithm depend more on the extent of the degradation. The requirement is modest for small localized degradations.

IV. Recent Results

A comparison between the projection algorithm and the SVD was also conducted for a one-dimensional case. This problem was worked previously when the projection algorithm was compared with the least-squares filter. In order to easily implement the SVD, the problem was reduced in size.

The original signal (Fig. 1) consists of two rectangular pulses of unit magnitude. The discretized pulses have five points each, separated by two zero points. The degradation is effected in the Fourier Transform domain. The Discrete Fourier Transform (DFT) of the signal is obtained by a Fast Fourier Transform routine. A triangular low-pass filtering operation (with cutoff frequency equal to .11 times the maximum frequency) yields the degraded signal (Fig. 2). Note that in the equation $g = [H] f$, $M = 64$ and $N = 116$ because the values at the edges of the degraded signal are in part determined by values beyond the edges. So there are more unknowns than equations.

The additive noise was Gaussian with standard deviation σ. Two values of σ were used in the study. The value $\sigma = 0.05$ yields a signal-to-noise ratio of $20 \log_{10} \frac{1}{.05} = 26$ dB. The degraded signal plus noise is shown in Fig. 3. The projection method results are given in Fig. 4a, b, c,. The values of the signal were restricted to positive values after each iteration. The results without this constraint were almost as good. The pulses are clearly resolved into two peaks after 30 iterations. No significant improvement occurred after this point.

The SVD results for $\sigma = 0.05$ are given in Fig. 5a, b, c, d, e. The best restoration occurs after 15 terms. The projection algorithm gives clearly superior results for this case.

The amount of noise was increased to $\sigma = 0.075$, a signal-to-noise ratio of $20 \log_{10} \frac{1}{.075} = 22.5$ dB. The degraded signal plus noise is shown in Fig. 6. The projection results are given in Fig. 7a, b, c. It should be noted that the degraded signal plus noise was always chosen to be the initial guess solution in the iterative process. The SVD restoration is presented in Fig. 8a, b, c, d. The projection algorithm results are even more clearly superior for this case.

The one-dimensional study shows that the projection algorithm yields better results than the SVD in this case. It is not certain why the SVD seems to perform better in the two-dimensional case. One reason might be that the SVD restorations are much smoother. The projection method does a better job of resolving two pulses, but in a two-dimensional image, the restoration is less smoothly varying in nature. Consequently, the SVD results are more pleasing to the eye.

The time required to generate and store the eigenvectors and eigenvalues on magnetic tape was 146 secs. More time is needed to read this data from the tape. If all the needed values are in the central memory, however, the SVD will calculate the restoration faster than the projection algorithm. The SVD uses .007 secs. to calculate one term. So 15 terms requires .105 secs. The projection algorithm takes 0.21 secs. per iteration, which is 6.3 secs. for 30 iterations.

V. Future Work

There are several studies which can be conducted on the projection algorithm. The method is now being applied to actual satellite data. For this data, it is necessary to interpolate between the picture elements to achieve restoration. With the projection method, it is hoped that the improvement in resolution will be helpful in locating edges and thus increase the accuracy of area measurements. If the data is to be used in a classification experiment, the method may increase the accuracy of classification.

Further studies will include: the effect of noise on the convergence of the method, the effect of errors in the matrix [H], efficient storage of the elements of [H], analysis of the case of inconsistent equations, theoretical justification of the positivity constraint, and the performance of the method for more complex types of degradation.

REFERENCES

1. T. S. Huang and P. M. Narendra, "Image Restoration by Singular Value Decomposition," Applied Optics, Vol. 14, pp. 2213-2216, Sept. 1975.

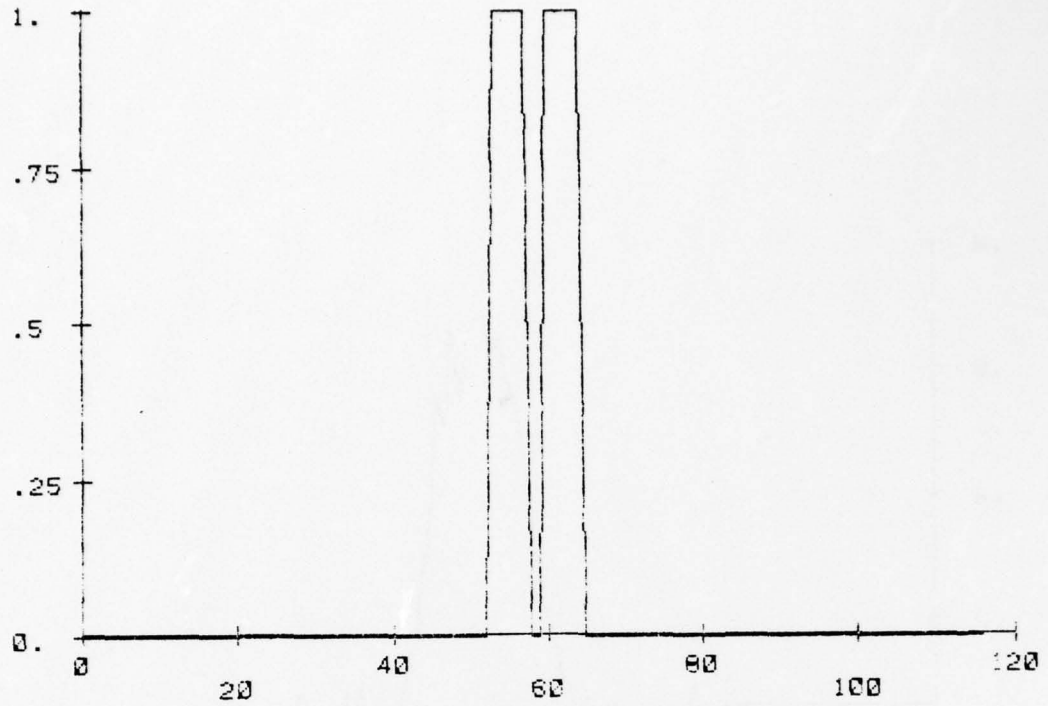


Figure 1 Original signal

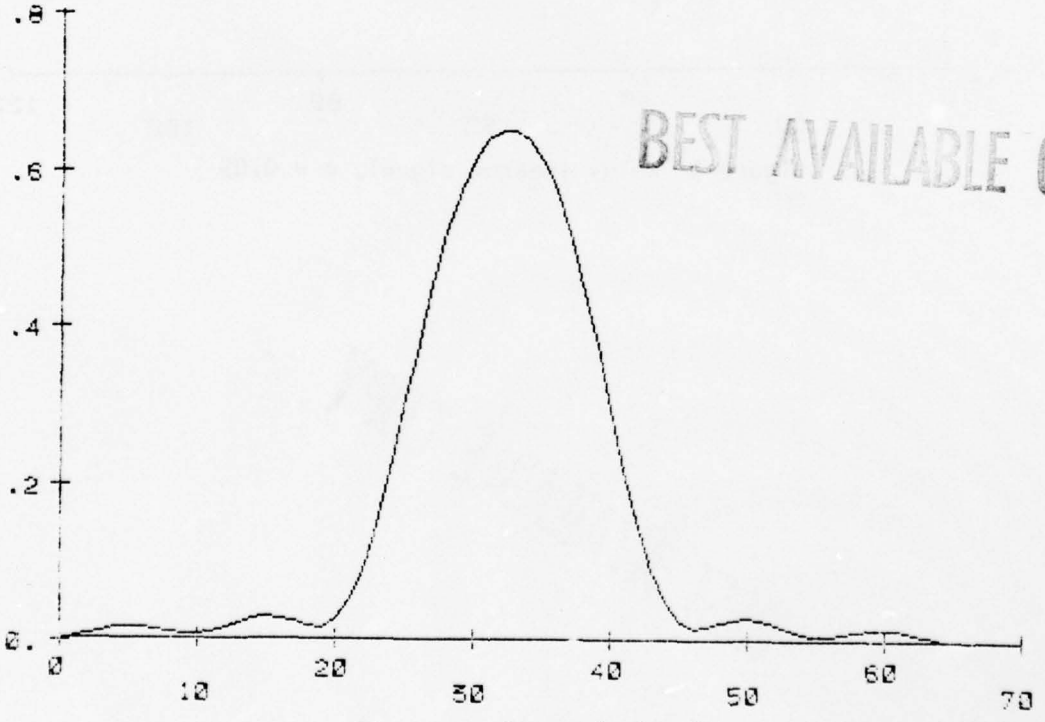


Figure 2 Smeared signal

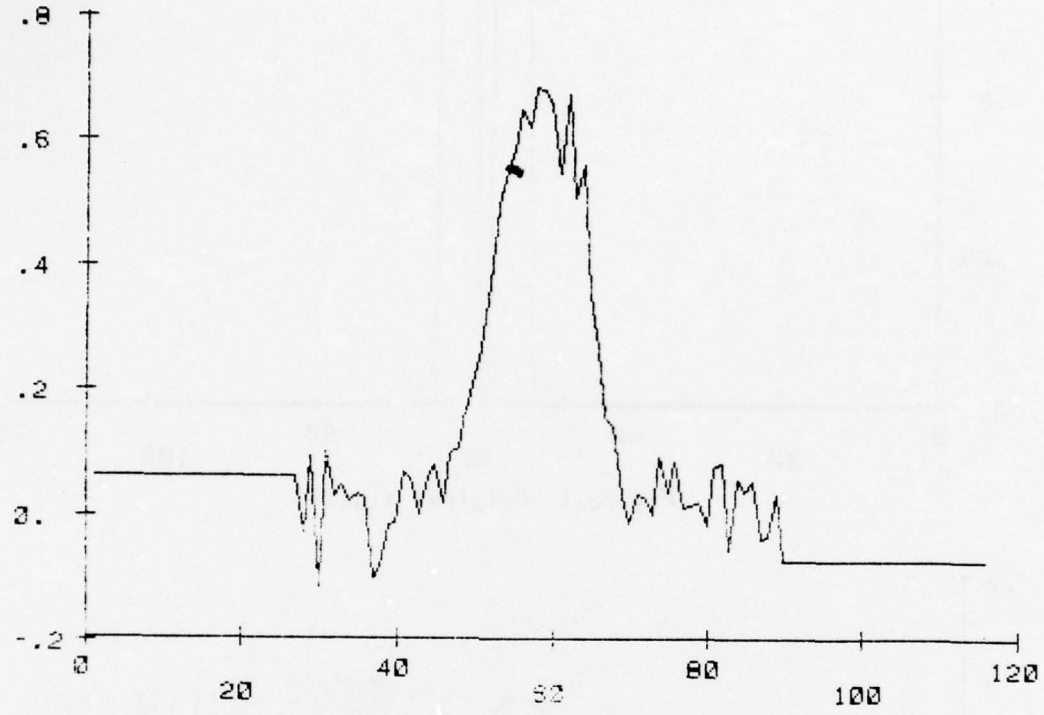


Figure 3 Noisy smeared signal, $\sigma = 0.05$

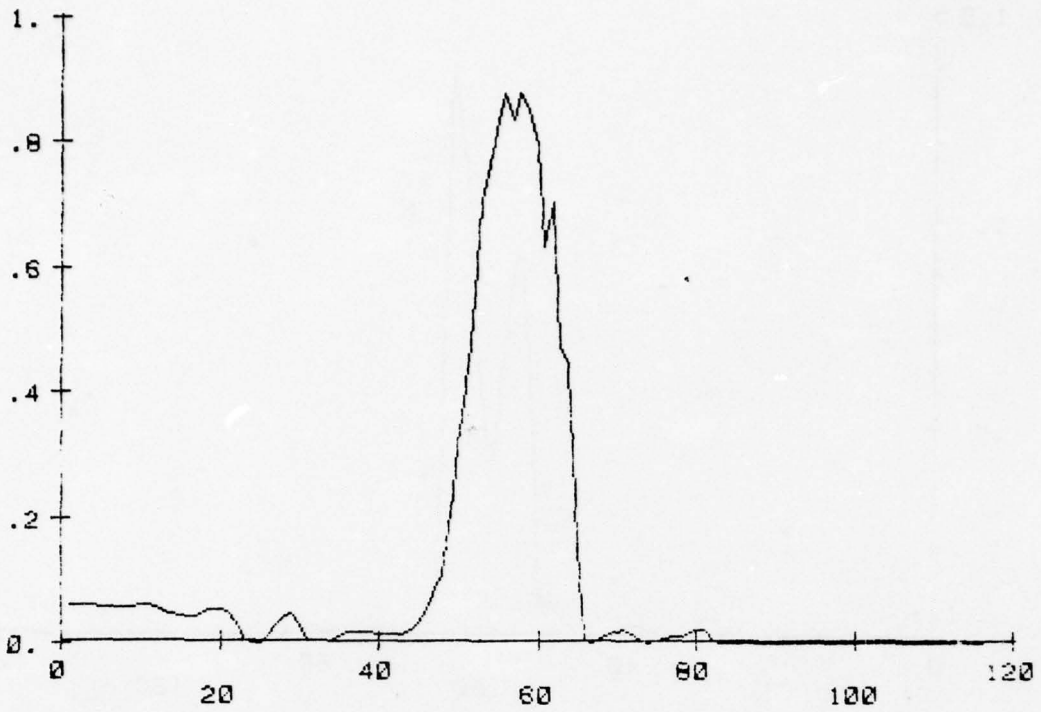


Figure 4a Iterative restoration, positivity constraint (1 iteration)

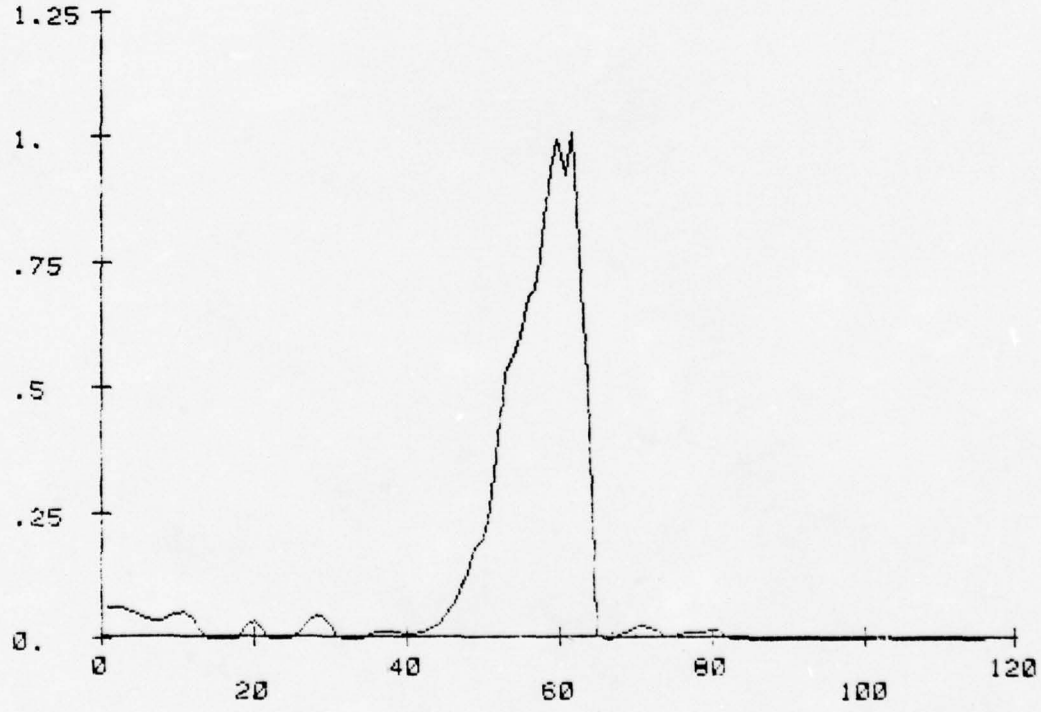


Figure 4b Iterative restoration, positivity constraint (5 iterations)

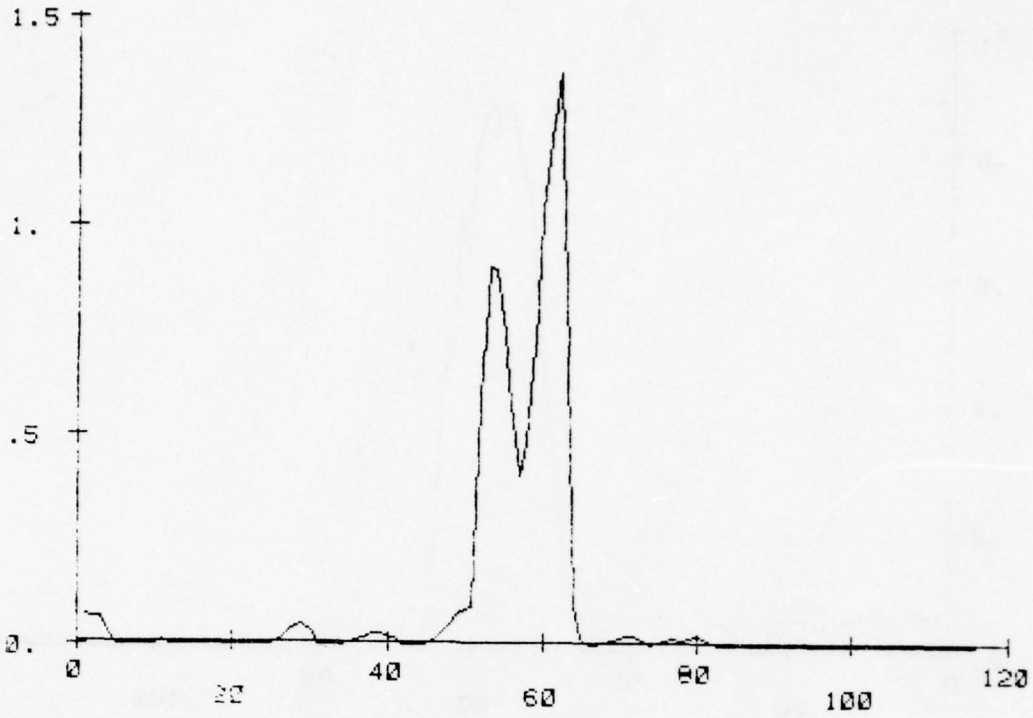


Figure 4c Iterative restoration, positivity constraint (30 iterations)

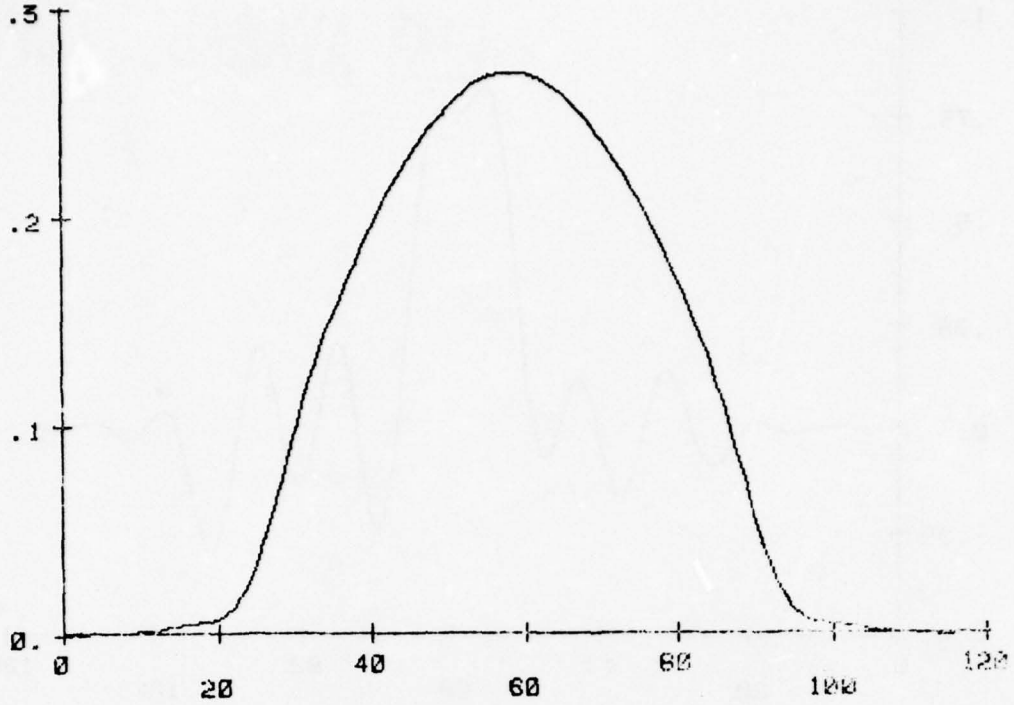


Figure 5a SVD (1 term)

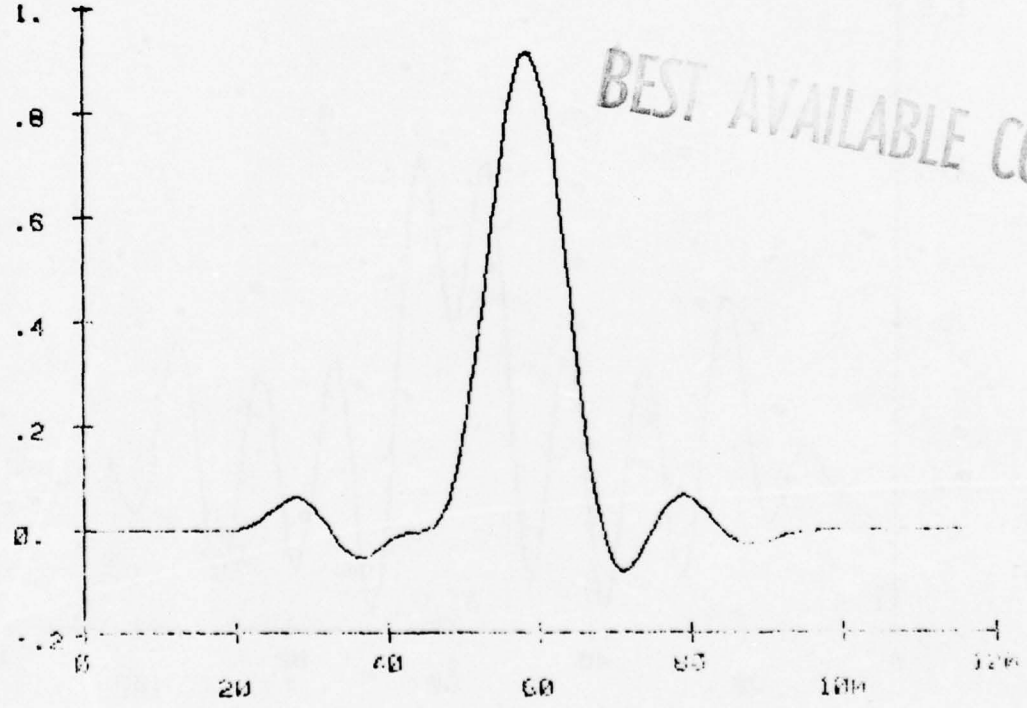


Figure 5b SVD (9 terms)

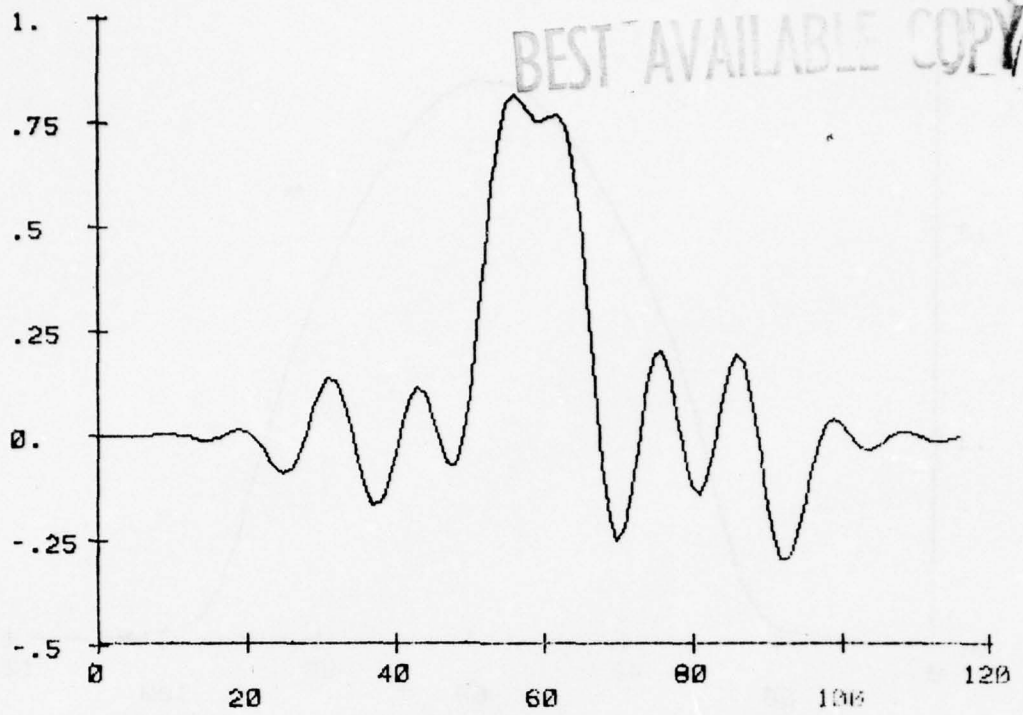


Figure 5c SVD (14 terms)

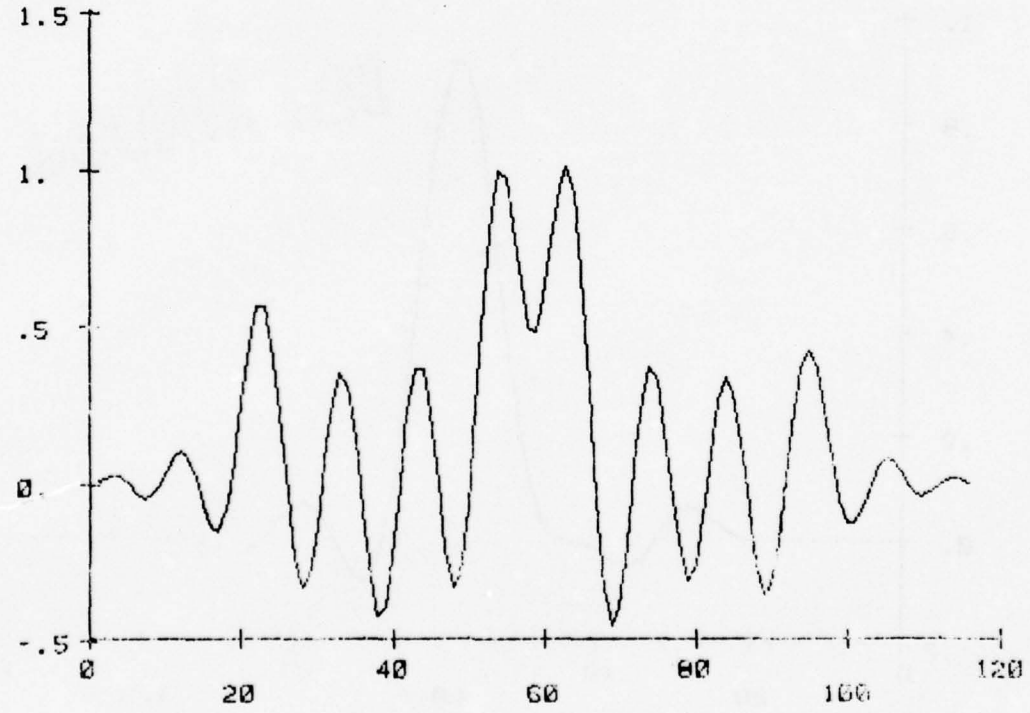


Figure 5d SVD (15 terms)

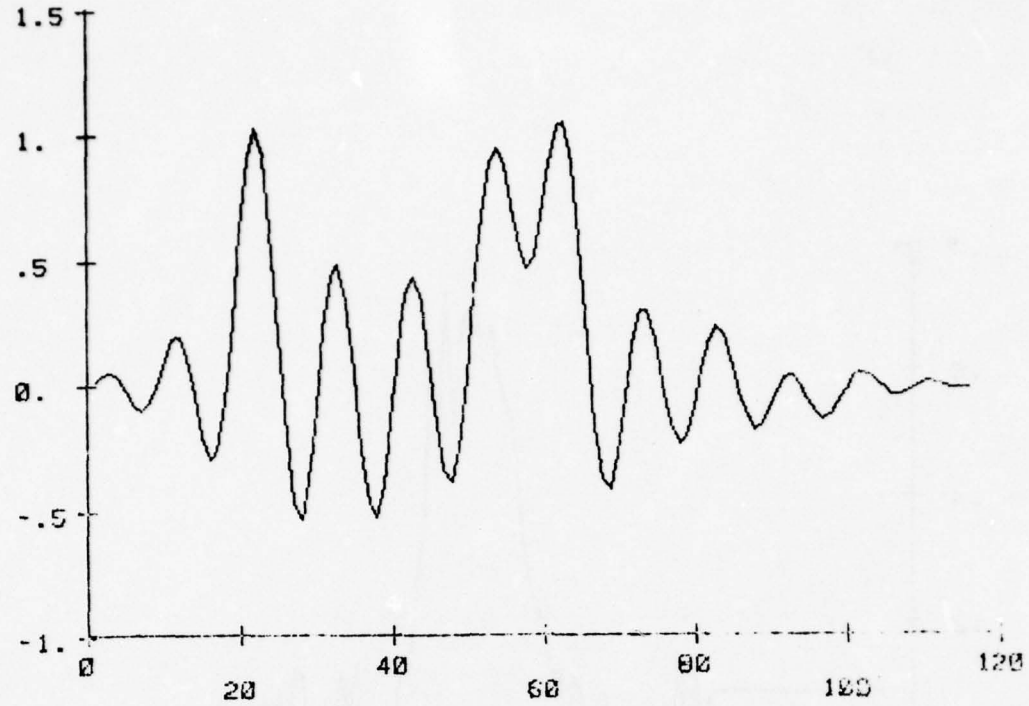


Figure 5e SVD (16 terms)

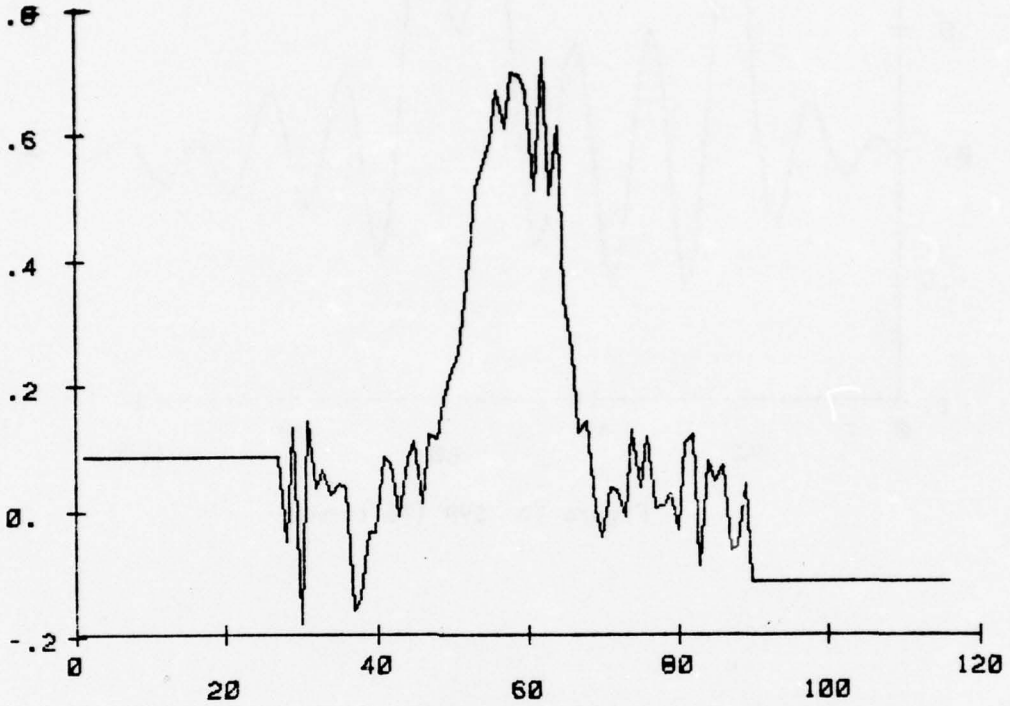


Figure 6 Noisy smeared signal, $\sigma = 0.075$

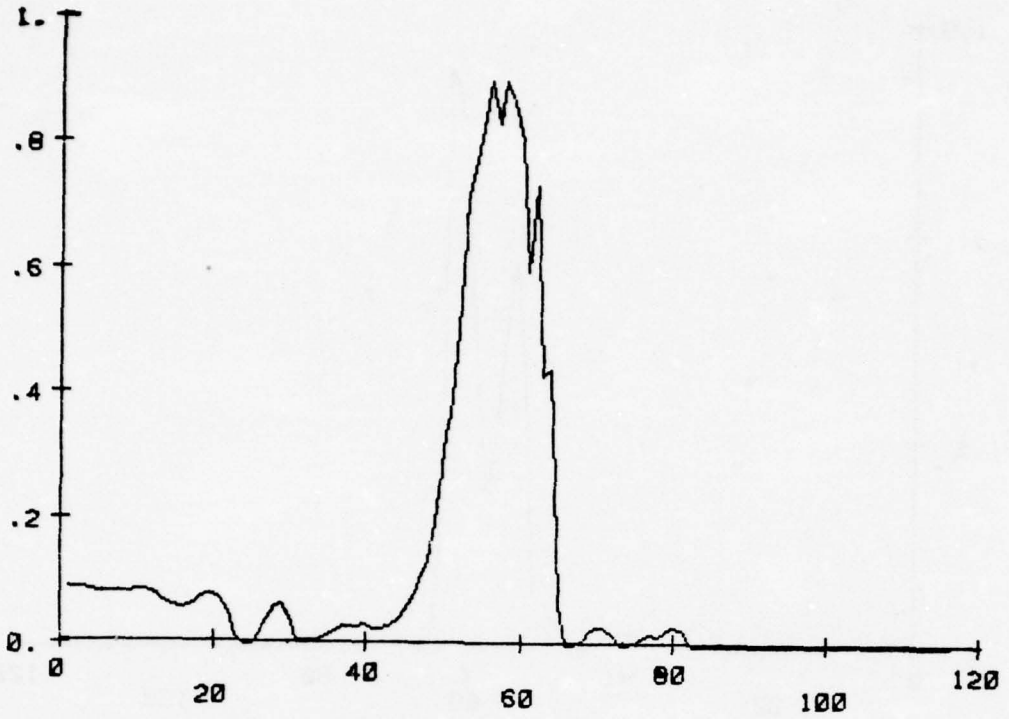


Figure 7a Iterative restoration, positivity constraint (1 iteration)

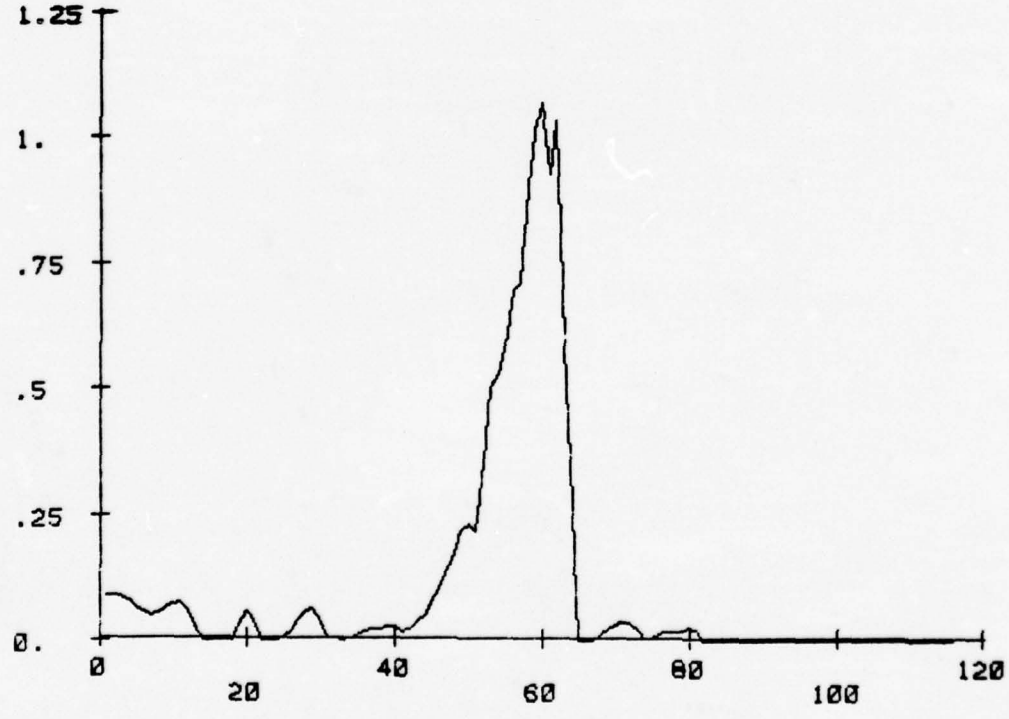


Figure 7b Iterative restoration, positivity constraint (5 iterations)

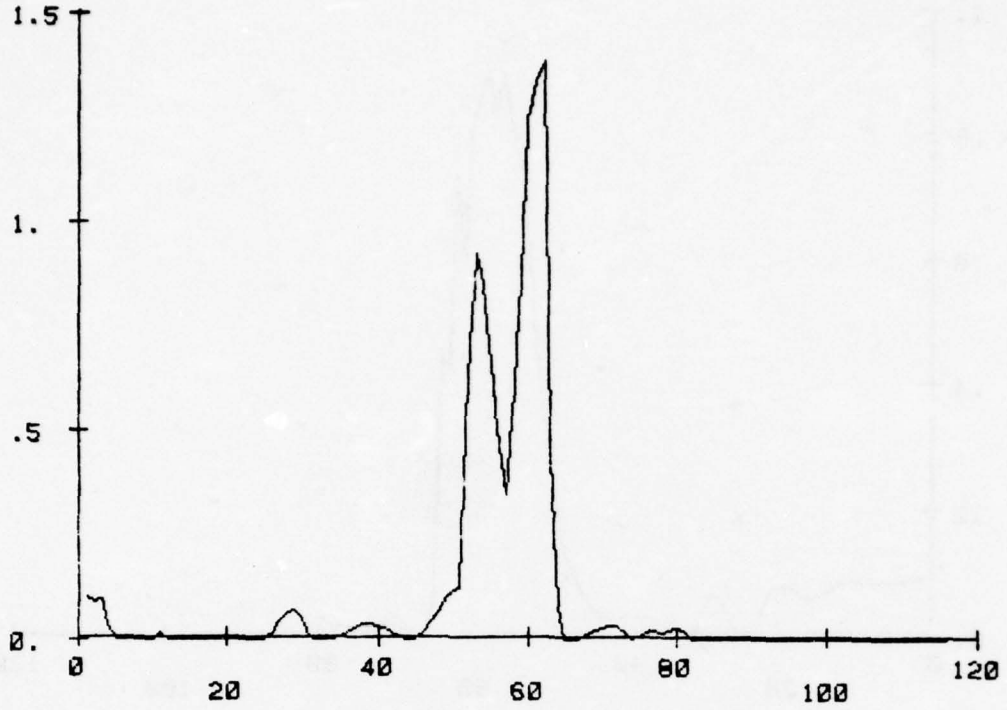


Figure 7c Iterative restoration, positivity constraint (30 iterations)

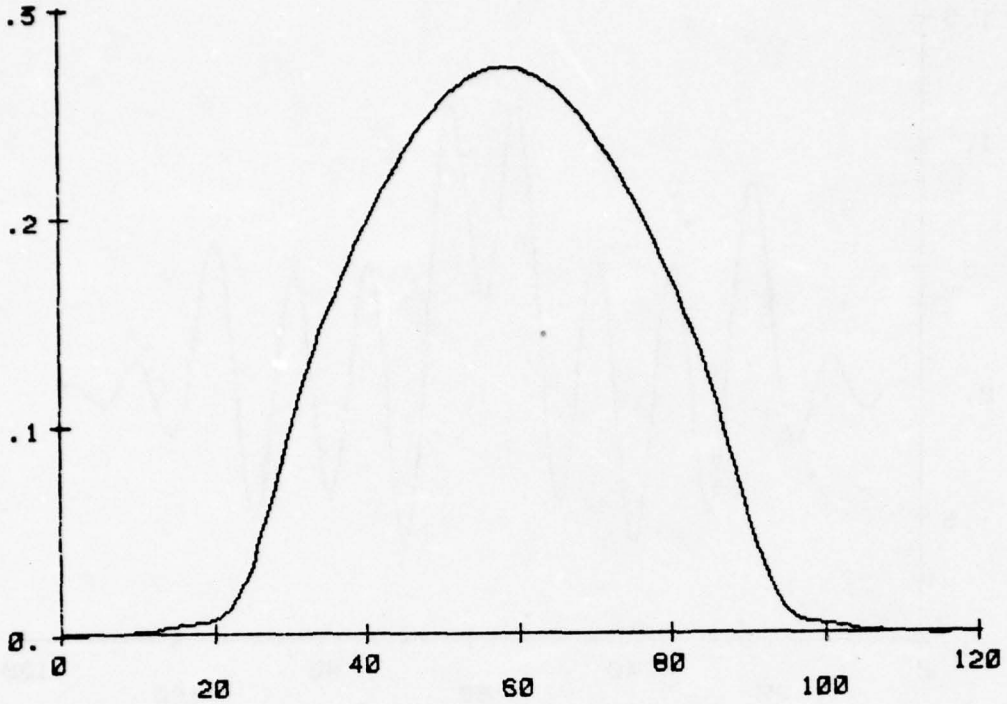


Figure 8a SVD (1 term)

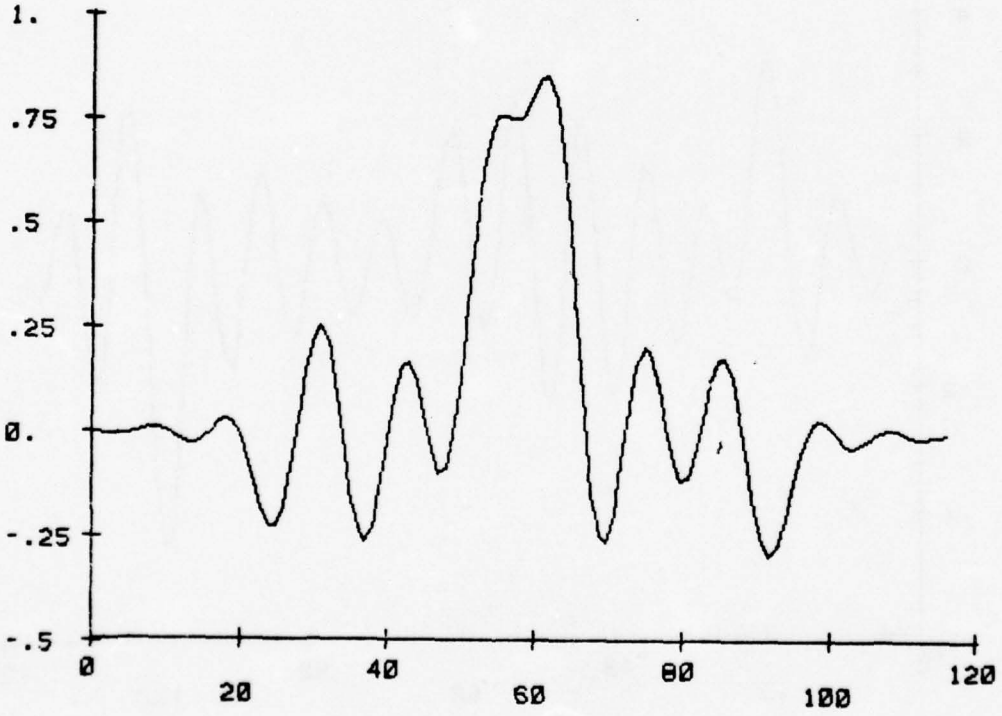


Figure 8b SVD (13 terms)

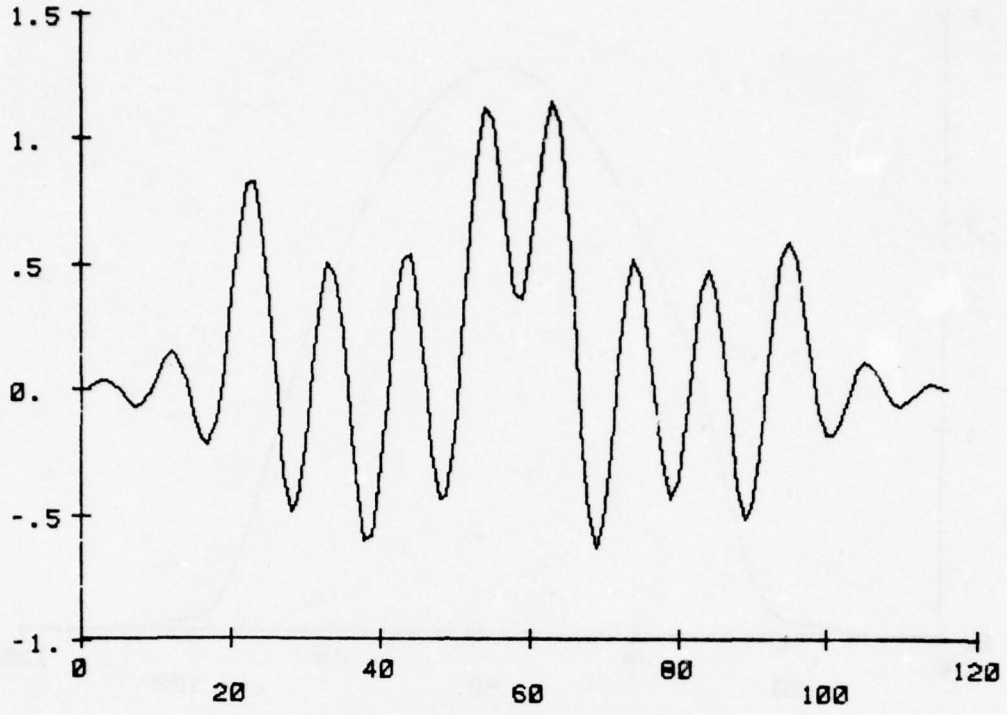


Figure 8c SVD (15 terms)

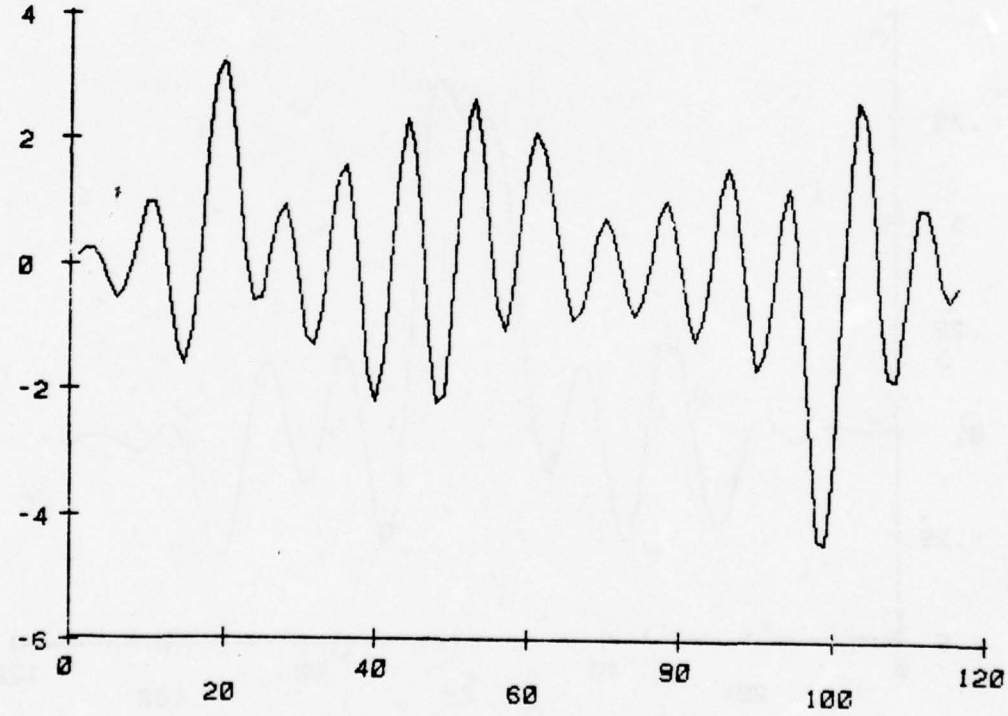


Figure 8d SVD (19 terms)

FOURIER DESCRIPTORS FOR EXTRACTION OF SHAPE INFORMATION

T. Wallace and P.A. Wintz

FOURIER DESCRIPTORS

The Fourier descriptor (FD) is one method of describing the shape of a closed, planar figure. Given a figure in the complex plane, the contour can be traced, yielding a (one-dimensional) complex function of time. If the contour is traced repeatedly, the periodic function which results can be expressed in a Fourier series. Granlund [1] defines the FD of a contour as the coefficients of this Fourier series.

To implement this method of shape description, it is necessary to sample the contour at a finite number of points. Since the discrete Fourier transform of a sequence gives us the values of the Fourier series coefficients of the sequence, assuming it to be periodic, using an FFT algorithm satisfies the definition above. The computational advantages of the FFT are well known.

Once the Fourier descriptor has been computed, the operations of rotation, scaling, and moving the starting point are easily implemented in the frequency domain by simple arithmetic on the frequency domain coefficients. While shapes may be compared in the space domain, the procedures required to adjust their size and orientation are computationally very expensive. Normally an iterative type of algorithm is employed, which searches for an optimum match between the unknown shape and the test set.

This Fourier descriptor algorithm normalizes the size and orientation of a shape before any comparisons to test shapes are made. The classification process becomes a simple clustering problem with no iterative searches to contend with.

Granlund's approach to shape information extraction involves defining "Fourier descriptors" by considering products of Fourier series coefficients

which are shown to be invariant to position, size, orientation, and starting point factors. This results in an increase in data dimensionality from N to $N^2/2$, without any change in total information. (Since the FFT is a reversible linear transformation, all the shape information is contained in the original N coefficients.) Our approach is to work with the original FFT output vector, normalizing it to a standard position, size, orientation, and starting point, so that classification algorithms become more efficient. The classification problem becomes more amenable to theoretical analysis, as well.

NORMALIZATION

The frequency domain operations which affect the position, size, orientation, and starting point of the contour follow directly from properties of the DFT. To change the position of a contour, just vary the zero frequency (DC) coefficient of the FD. Adding a complex constant to every point in the time domain representation of a contour is equivalent to adding that value to the DC term of the DFT.

To change the size of the contour, the components of the FD are simply multiplied by a constant. Due to linearity, the inverse transform will have its coordinates multiplied by the same constant.

To rotate the contour in the time domain simply requires multiplying each coordinate by $e^{j\theta}$ where θ is the angle of rotation. Again by linearity, the constant $e^{j\theta}$ has the same effect when the frequency domain coefficients are multiplied by it.

To see how the contour starting point can be moved in the frequency domain, recall the time shifting property of the DFT. Shifting the starting point of the contour in the time domain corresponds to multiplying the ith frequency coefficient in the frequency domain by $e^{ji\,T}$, where T is the fraction of a period through which the starting point is shifted. (AS T goes from 0 to 2π,

the starting point traverses the whole contour once.)

Given the FD of an arbitrary contour, the normalization procedure requires performing the normalization operations such that the contour has a standard size, orientation, and starting point. The following method of FD normalization preserves all of the shape information while rejecting noise effectively. In order to reject noise, the coefficients used in the procedure are chosen to have as large magnitudes as possible.

First, we require the phases of the two largest coefficients to be zero. A(1) will always be the largest, with magnitude unity due to the scale normalization procedure which defines that magnitude. Let the second largest coefficient be A(k). (The frequencies of the coefficients produced by an FFT of length n range from $-(n/2)+1$ to $(n/2)$). The normalization multiplicity m of coefficient A(k) is defined as:

$$m = |k-1|$$

THM: The requirement that A(1) and A(k) have zero phase angle can be satisfied by m different orientation/starting point combinations.

PROOF: Use the two allowable operations to arrive at one orientation and starting point which gives zero phase for A(1) and A(k). Next use the starting point movement operation (multiplication of the ith coefficient by e^{jiT}) to move the starting point once around the entire contour. To accomplish this T must range from 0 to 2π. Now consider the two cases k positive and k negative. If k is positive, the phases of A(1) and A(k) will coincide at k-1 different starting points. But at each of these starting points, we can use the orientation operation (multiplication of each coefficient by $e^{j\theta}$) to reduce the phases to zero. Similarly, if k is negative, the phases of A(1) and A(k) will coincide at 1-k different starting points. Again, the orientation operation can reduce the phases to zero.

Note that if k=2, the orientation and starting point are defined uniquely. In general, however, A(2) will not be the second largest coefficient in magnitude so this ambiguity must be resolved to achieve a general procedure.

The obvious method of solving this problem is to check the phase of a third coefficient A(p) at each of the m possible orientation/starting point combinations and choose the normalization which gives a phase closest to zero for this coefficient. However, this ambiguity-resolving coefficient cannot be chosen arbitrarily. If the normalization multiplicity of coefficient A(p) is the same as that of A(k), or a multiple of it, the phase of A(p) will be the same at each possible normalization! If m for coefficient A(p) (denoted m[p]) is a factor of m[k], or a multiple of a factor of m[k] less than m[k], there is also ambiguity since some of the m possible normalizations will result in identical phases for A(p). If these ambiguous coefficients are removed from consideration, and the unambiguous coefficient with the largest magnitude is used to select one of the m allowable normalizations, a general procedure is obtained.

To breifly review the entire normalization procedure, we start by dividing each coefficient by the magnitude of A(1) to normalize the size of the contour. We find the coefficient of second largest magnitude and compute its normalization multiplicity. We then locate the third largest coefficient suitable for resolving the ambiguity (A(p)) as explained above. The orientation and starting point are adjusted to satisfy the restrictions that A(1) and A(k) are real and positive, and A(p) has phase as close to zero as possible.

This method is quite powerful, but a slight modification in the procedure has been found helpful in those cases in which there are two or more coefficients suitable for use as A(p) with almost the same magnitude. It is very unlikely that the magnitudes will be identical, but if they are even close, noise may

cause one of them to be used to normalize the test FD, and the other to normalize the unknown FD. To overcome this, the ambiguity resolving coefficient used to normalize the test FD can be supplied to the normalization subroutine directly, rather than having the subroutine compute it.

PRACTICAL CONSIDERATIONS

Theoretically, the procedure involves an exact representation of a contour which is sampled at uniform spacing. While nonuniform spacing can result in a frequency domain representation which converges faster, there are obviously great difficulties involved in attempting to define a standard sampling strategy using non-uniform spacing.

Remembering that the FFT algorithm requires an input vector whose length is a power of 2, it is clear that the length of an arbitrary chain code representation must be adjusted before the FFT can be used. The obvious procedure for doing this is to compute the perimeter of the contour, divide it by the desired length (desired power of 2), and starting at one point, trace around the contour saving the coordinates of appropriately spaced points. The desired power of 2 might be the smallest power of 2 larger than the length of the chain code.

Practically, the input to the shape analysis algorithm will be a contour taken from a sampled picture. The perimeter of this contour will be a chain code approximation to the actual perimeter of the contour. While it can be argued that for high enough sampling density in the original picture, the chain code is an arbitrarily good approximation to the contour, this argument breaks down if you consider the density of points around the approximate contour versus the exact contour.

Consider an equilateral right triangle oriented so that the legs line up with the x and y axes, with the hypotenuse at 45 degrees. The "length" of the

contour, if an ordinary four neighbor chain code is used, will be four times the length of one leg; the hypotenuse will be as long as both legs combined! Obviously the density of points on the hypotenuse will depart from the proper value by a factor of $\sqrt{2}$. This error will cause the normalized Fourier descriptors (NFD's) of simple figures such as triangles to differ substantially, and render the algorithm virtually useless.

One solution to this problem is to use an eight neighbor chain code, in which the four diagonal neighbors of a point can also be the next point in the chain. In the example just considered, this eliminates the point density error. Of course, for different orientations, there will still be a certain amount of error due to the chain code approximation, but this is reduced from a maximum of about 40% to a maximum of about 8%. Experimental results using the eight neighbor chain code confirm that this error is tolerable. If a picture is contoured using a four neighbor chain code and it is desired to process the contours using the FD method, the four chain codes can be easily converted to approximate eight chain codes which are suitable for analysis.

CLASSIFICATION METHODS

Given two NFDs, how do we measure their degree of similarity? An appropriate classification method is essential if we are to compare unknown shapes to a test set.

Consider two sampled contours a(i) and b(i), and define the difference c(i) = a(i) - b(i). Evidently if a(i) and b(i) are identical, c(i) is identically zero. If a(i) and b(i) are not identical, the magnitudes of the c(i) coefficients are a reasonable measure of the difference between a(i) and b(i). Now consider the frequency domain vectors corresponding to a(i), b(i), and c(i), denoted A(i), B(i), and C(i). Due to linearity, we have C(i) = A(i) - B(i). Applying Parseval's theorem to the difference vector, we have

$$\sum_{i=0}^{N-1} c^2(i) = \frac{1}{N} \sum_{i=0}^{N-1} C^2(i)$$

$$\sum_{i=0}^{N-1} (a(i) - b(i))^2 = \frac{1}{N} \sum_{i=0}^{N-1} (A(i) - B(i)^2$$

In other words, the sum of the squares of the differences of the real and imaginary parts of each coefficient of two FDs is proportional to their point by point mean square error in the space domain. The mean square distance measure in the frequency domain is seen to correspond to a reasonable time domain criterion which weights each point equally. In recognizing a contour corrupted by such factors as quantization error or poor photographic resolution, such a criterion seems appropriate. The effectiveness of this classification method is demonstrated by the experiments described below.

The only other classification algorithm investigated used an absolute value of the difference between the real and imaginary parts of each FD coefficient. While the exact time domain equivalent of this operation is mathematically intractable, a few qualitative observations can be made. The absolute value criterion is more tolerant of a large difference involving a single component or two than is the mean square criterion. The coefficients with the greatest expected magnitudes are those of lowest frequency. If one of these coefficients is changed, the effect is generally to distort the contour globally, such as stretching it out in one direction. Hence, this distance measure might be useful in identifying contours which are very similar in smaller detail, but whose point by point difference might be substantial. It is possible that this may correspond to similarity as defined by human observers more closely than does the mean square distance. We would not expect, however, that this method would be as effective as the mean square method for matching

a test contour to a representation of that exact contour corrupted by quantization noise or poor resolution.

Since the closed contour is a continuous function, the Fourier series converges fairly rapidly, as would be expected. Most of the M.S. distance between two FDs is due to relatively few coefficients, and the classifications reported below use no more than 32 coefficients.

FD - CONTOUR RELATIONSHIPS

If a FD consists of coefficient A(1) only, with all other coefficients zero, it will transform back to the time domain as a sampled circle. Higher frequency coefficients also transform back as sampled circles, but they transverse the circle a number of times. A(k) will yield a time domain sequence which traces a sampled circle k times in the counterclockwise direction. A(k) and A(-k) together yield a sampled ellipse, in a manner analagous to the elliptical polarization of electromagnetic theory.

Due to linearity, a contour in the time domain consists of a sum of the inverse transforms of its FD coefficients. Hence this view of each FD coefficient as a sampled "phasor" yields insight into the relationships between a contour and its FD. A(1) is the fundamental frequency coefficient which is always the largest in magnitude, and is forced to have magnitude unity by the magnitude normalization procedure. It is of interest to describe the figures generated by A(1) and A(k) combined, with all other coefficients zero, since often most of the "energy" of a FD is contained in as few as two coefficients. Interestingly enough, the "normalization multiplicity" m defined above plays a part here, with the contour resulting from nonzero A(1) and A(k) having m[k]-fold rotational symmetry. Granlund observed that contours with k-fold rotational symmetry consist of components whose frequencies are multiples of k-1. If k is negative, the contours are similar to polygons, and if k is

positive, the contours generally are quite round, and appear to be loops superimposed on a circle. If k=-1, the contour is of course a sampled ellipse. Most contours of interest taken from actual photographic data have a negative frequency coefficient as the second largest in magnitude.

Figure 1 shows four contours whose FDs have only two nonzero coefficients. The magnitudes of two coefficients completely determine the shape of the figure generated, with the phases only affecting orientation and starting point. Note also that the uniform sampling condition in the time domain is not satisfied when any arbitrary FD is inverse transformed.

Consider now a bilaterally symmetric contour in the time domain.

THM: A Fourier Descriptor represents a bilaterally symmetric contour iff the rotation and starting point shift operations can be performed such that the imaginary part of each FD coefficient (except A(0)) is zero.

PROOF:

only if

Assume a sampled contour a(i) has bilateral symmetry. Since translations do not affect the shape of the contour, without loss of generality, we can move the contour to the origin so that coefficient A(0) = 0. Perform the rotation and starting point operations so that

a) the starting point is on the axis of symmetry

(two points will satisfy this)

b) the starting point is on the real axis

The real axis now coincides with the axis of symmetry of the contour. Since the vectors of the DFT matrix are linearly independent, no linear combination of other coefficients can cancel the effect of a given coefficient. Hence the successive approximations to the original contour obtained by adding in the FD coefficients one at a time must all have bilateral symmetry, and their

axes of symmetry must be the real axis. Consider first the partial contour consisting of the inverse DFT of A(1) only. In order for the starting point of this partial contour to be on the real axis, A(1) must be real, since the first point of the inverse DFT of a sequence is merely the sum of all of the terms in the sequence. Now consider the nth approximation to the given contour. Assume that these n coefficients are real. Adding the (n+1)th coefficient must keep the sum of n+1 coefficients real, so the (n+1)th coefficient must be real. By induction, all the FD coefficients must be real.

if

Now assume that a FD is given consisting of only real coefficients. Let A(0) = 0. Consider again the sequence of successive approximations to the time domain contour obtained by adding one FD coefficient at a time. The a(i) resulting from A(1) alone evidently have bilateral symmetry, with one axis of symmetry coinciding with the real axis. Now consider adding an arbitrary A(k) to A(1) and inverse transforming this vector of two nonzero coefficients. In order for this contour to have bilateral symmetry about the real axis, we require $a(i) = a(-i)^*$. (* denotes complex conjugation) (recall that both the a(i) and A(i) sequences are periodic) from the definition of the IDFT,

$$a(i) = \frac{1}{N} \left[A(1)\, e^{ij2\pi/N} + A(k)\, e^{ij2\pi k/N} \right]$$

$$a(-i) = \frac{1}{N} \left[A(1)\, e^{-ij2\pi/N} + A(k)\, e^{-ij2\pi k/N} \right]$$

where N is the length of the DFT. Since the A(i) are real, these quantities are evidently complex conjugates, so the partial contour has the desired symmetry. Now adding each additional coefficient will add a component to the contour which possesses this symmetry. Hence the contour is bilaterally

symmetric, and in fact the axis of symmetry is the real axis, and the starting point is on the axis of symmetry.

AIRCRAFT RECOGNITION

This method of extracting shape information was experimentally tested on 20 airplane silhouettes which were digitized to two different resolutions. The high resolution versions were quite accurate representations of the aircraft, while the low resolution versions showed significant distortion of some of the smaller features such as engines. Using the high resolution contours as a test set, an attempt was made to classify the low resolution contours using this FD algorithm. Using a mean square distance measure, 95% classification accuracy was attained. The aircraft were of four different types. Figures 2 and 3 show high and low resolution contours representing each type. Figure 4 shows the magnitudes of the NFDs computed from the high resolution contours.

The aircraft outlines are approximately bilaterally symmetric, although quantization error prevents them from being exactly symmetric. The normalization procedure always will yield a NFD whose inverse transform has starting point on the real axis, and whose axis of symmetry coincides with the real axis, given the FD of a bilaterally symmetric contour. Which of the two points at which the axis of symmetry intersects the contour will actually be the starting point depends on the ambiguity-resolving procedure described above. The procedure generally favors the point furthest from the origin of the complex plane, but supplying a selected ambiguity-resolving coefficient to the normalization subroutine can reverse this. In case both possible starting points are approximately equidistant from the origin, the starting point resulting from normalization is somewhat unpredictable. This is the situation in which it is advisable to check that the unknown FD is normalized

using the same ambiguity-resolving coefficient as the test FD.

Since the actual experimental contours investigated were not perfectly bilaterally symmetric, the normalization subroutine did not always result in a starting point which falls on the best estimate of the axis of symmetry. However, since the algorithm was written to reject noise, the starting point was always quite close to the axis of symmetry.

THE BLOB ALGORITHM

The goal of this research is to extract shape information from actual photographic data. In order to obtain a chain code contour with which to apply the Fourier Descriptor classification technique, we must first obtain that contour from the original data. Present work in this area centers around the BLOB algorithm of Gupta and Wintz [2], [3]. The original BLOB program merged statistically similar pixels row by row, which is advantageous for processing purposes, but has the disadvantage of creating artificial contours in the vertical direction. Gupta and Wintz suggest that the contour tracing algorithm of Wilkins and Wintz [4] could be used to eliminate this problem. This "contour-tracing BLOB" is the version that we are presently working with, and all references to "BLOB" below refer to this version. The false contour problem has been eliminated, and the output of the program consists of initial point locations and grey levels, along with chain code directionals which are easily interfaced to the Fourier Descriptor programs.

Our problem in interfacing the BLOB directionals to the FD programs involves the fact that BLOB outputs 4-neighbor chain codes. A subroutine which converts these 4-codes to approximate 8-codes by simply removing all right angles and replacing them with the appropriate diagonal has been successfully used to overcome this. It might be preferable to rewrite BLOB to find 8-code contours directly, but it is felt that the classification improvement would

be small, and the computation time for BLOB would approximately double.

Another problem in analyzing BLOB output using the FD method concerns "dead ends". The contour tracing algorithm occasionally finds a narrow channel which is statistically similar to the present contour, but which ends in a dead end. When this happens, BLOB retraces its path back to the original area, and continues to construct the contour. This "dead end" path has a major effect on the FD of the contour, but it is probably not part of the main shape we are looking for. Hence these "dead ends" are removed by another subroutine before the FDs are computed.

The original BLOB program was used to process multispectral data, which was assumed distributed normally. The approach used was to first estimate the variance, which is distributed with the f-distribution, and then to estimate the mean, which has the student's t distribution. In testing a new pixel group for possible inclusion into a "BLOB", the f-test tested for similarity of variances, and then the t-test made use of the variance information and tested for similarity of mean.

The normal assumption is quite successful in dealing with multispectral data, but it is ineffective in describing ordinary photographic data. The problem immediately encountered was failure to locate objects of low variance. If a region of a photograph is of almost constant grey level, the estimated variance will be very small. Since the mean test involves a set number of standard deviations, the total range of means allowed to pass the mean test could be prohibitively small. Our present solution to this problem involves assuming that we have apriori information that the variance is at least some minimum value. This prevents the program from ever estimating a variance to be zero, for example, and consequently breaking up a region of virtually constant grey level.

Figure 5 is an aerial photograph containing two airplanes. A reference contour for the aircraft shown was computed by simple thresholding techniques using the same data (Figure 7). (Recall that BLOB considers two by two pixels, and thus has only half the resolution which can be achieved from the original data) The BLOB program was run on the picture shown, and the output was analyzed by the FD program. Considering all contours of length 50 to 1024, the aircraft were successfully identified, using only shape information. If all the contours traced by BLOB and tested by the FD program were shown, they would virtually fill up the picture. Figure 6 shows only the contours which were classified as planes by the FD program. While size, average grey level, orientation, and position information was reported by the program, none of this information was used in the classification process. Figure 8 shows the output from the FD program run on the picture of Figure 5.

REFERENCES

[1] G. H. Granlund, "Fourier Preprocessing for Hand Print Character Recognition," IEEE Trans. on Computers, Vol. C-21, pp. 195-201, Feb. 1972.

[2] J. N. Gupta and P. A. Wintz, "Multi-Image Modeling," School of Electrical Engineering, Purdue University, West Lafayette, IN Tech. Rep. TR-EE 74-24, 1974.

[3] J. N. Gupta and P. A. Wintz, "A Boundary Finding Algorith and its Applications," IEEE Trans. Circuits Systems, Vol. CAS-22, April 1975.

[4] L. C. Wilkins and P. A. Wintz, "Studies on Data Compression, Part 1, Picture Coding by Contours," School of Electrical Engineering, Purdue University, West Lafayette, IN, Tech. Rep. TR-EE 70-17, 1970.

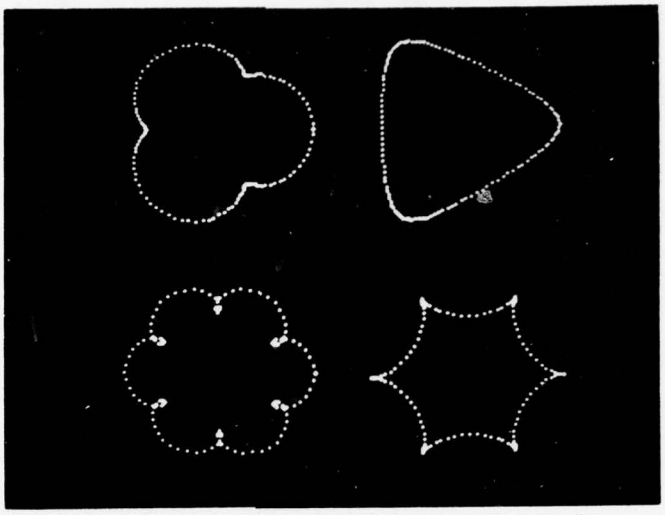


Fig. 1 Inverse Transforms of FD's Consisting of Selected Coefficients

Upper Left
A(1) = 1.0
A(4) = 0.2

Upper Right
A(1) = 1.0
A(-2) = 0.2

Lower Left
A(1) = 1.0
A(7) = 0.2

Lower Right
A(1) = 1.0
A(-5) = 0.2

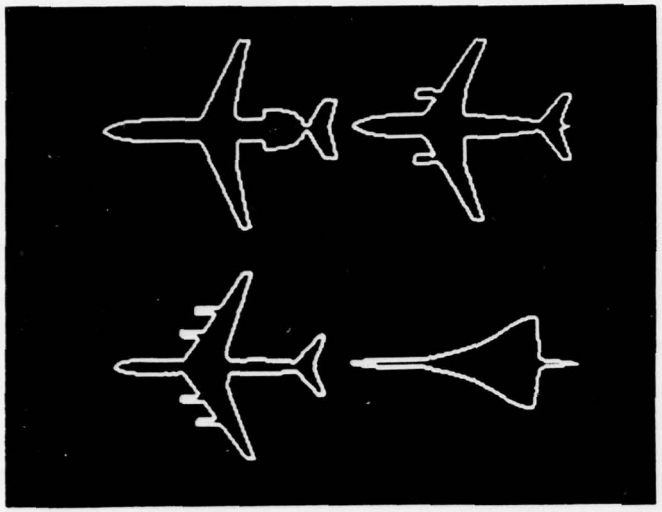


Fig. 2 Representative of High Res. Aircraft

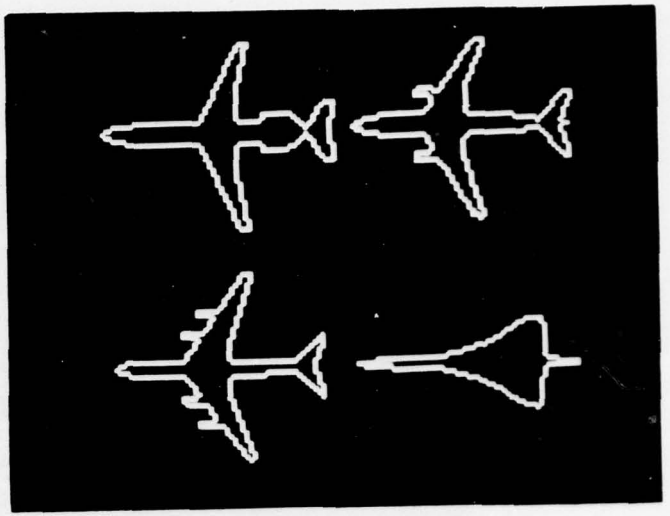


Fig. 3 Low Resolution Contours

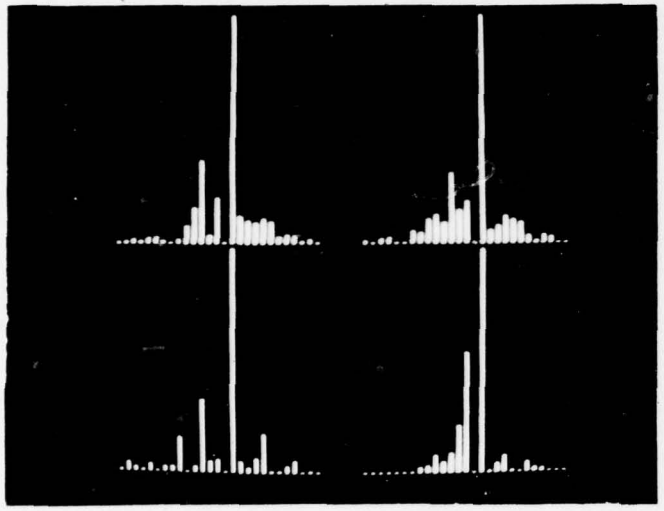


Fig. 4 Magnitudes of FD's Computed from Contours of Fig. 2

BAC	AIRBUS
111	A 300 B
DC 8	BAC
SERIES 50	CONCORDE

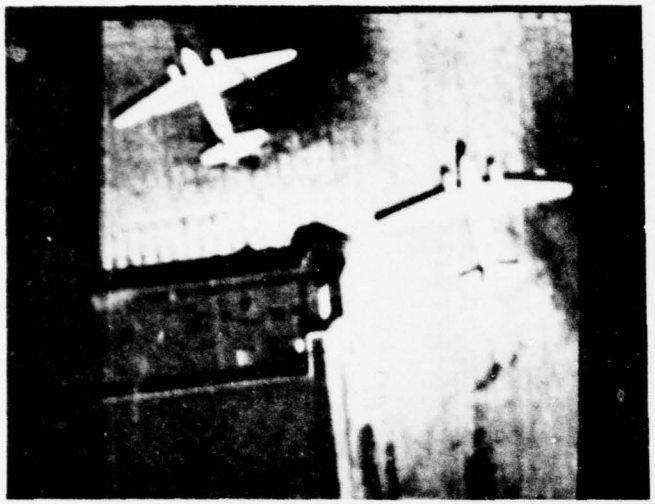


Fig. 5 Original Aerial Photograph

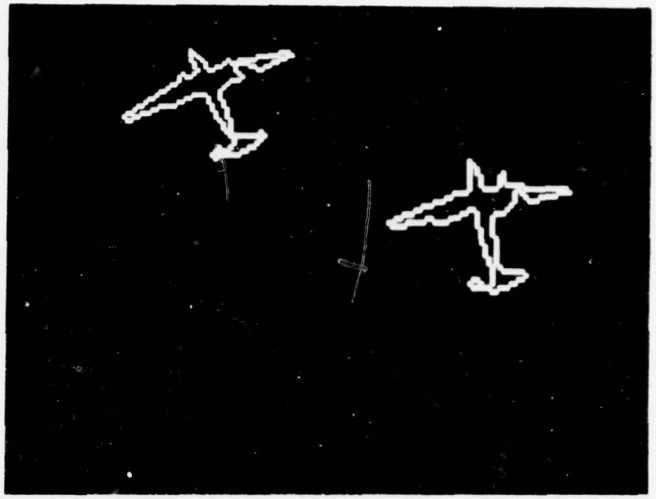


Fig. 6 Contours Found by BLOB Program

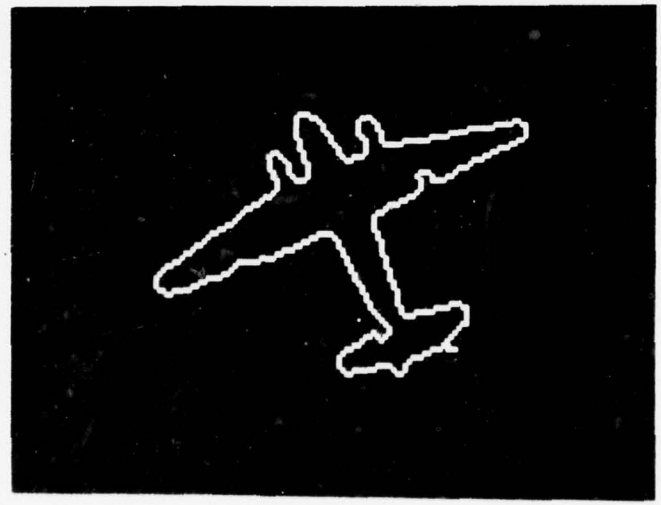


Fig. 7 Reference Contour Found by
Thresholding Techniques

Classification Results

Minimum length contour is 50
Maximum length contour is 1024

Classification Threshold is 0.10

Contour at 49 23 with mean 252.52
is classified as an airplane--distance is 0.0470
The picture contour is 12.14 times larger than standard size
The picture contour is rotated -1.16 radians.
The starting point is moved -3.11 radians.

Contour at 195 85 with mean 251.70
is classified as an airplane--distance is 0.0403
The picture contour is 12.29 times larger than standard size
The picture contour is rotated -1.38 radians.
The starting point is moved -3.04 radians.

Fig. 8

FILTERING TO REMOVE CLOUD COVER IN SATELLITE IMAGERY

O. R. Mitchell and E. J. Delp

I. INTRODUCTION

Satellite multispectral scans of the earth's surface such as those obtained from LANDSAT are often corrupted by cloud formations. The usual reaction is to discard these images as useless. However, in some situations, the data of interest is temporary and a clear scan of the area cannot be obtained. The question arises as to whether it is possible to filter out the cloud cover thus exposing the earth's surface below the clouds.

In order to investigate the potential of such a technique, a model of the cloud distortion process has been developed. The "noise" effects of the cloud are not a strictly additive or multiplicative process but a combination. Assuming the cloud cover is light enough so that some of the energy from the earth's surface passes through the cloud (in at least one spectral channel) a transformation can be developed which makes the signal and noise additive. Then optimum linear filtering techniques can be applied to separate the signal and noise. An appropriate inverse transformation then returns the filtered signal to the picture domain.

To apply this filtering technique an estimate of the signal or the noise statistics must be made. This is done by assuming the cloudy regions are generally brighter than the non-cloudy areas and that the clouds have relatively low spatial frequency content compared to the ground reflectance. Thresholds are set in both the picture and spatial frequency domains to allow an estimate of the noise statistics from the original cloudy image.

II. CLOUD DISTORTION MODEL AND FILTERING PROCEDURE

Assume that an image of the earth is produced when a light cloud cover exists over the region of interest as shown in Fig. 1. If we assume that the

cloud reflection of sunlight plus the cloud transmission equals one (ignoring diffusion) and that the sun illumination is approximately constant on the earth's surface, then the received image at the scanner is

$$s(x,y) = a\,L\,r(x,y)t(x,y) + L[1 - t(x,y)] \leq L \qquad (1)$$

where $r(x,y)$ is the image of interest and $t(x,y)$ is due to the clouds. The values of $r(x,y)$, $t(x,y)$, and "a" (sunlight attenuation) range between 0 and 1.

A transformation is now performed by subtracting $s(x,y)$ from L and taking the logarithm:

$$\log[L - s(x,y)] = \log[t(x,y)] + \log[L - a\,L\,r(x,y)] \qquad (2)$$

If the signal is now assumed to be $\log[L - a\,L\,r(x,y)]$ and the noise is assumed to be $\log[t(x,y)]$, then the signal and noise are additive and uncorrelated. Weiner linear filtering techniques can now be used to remove the noise [1,2].

This method of converting a multiplicative process to an additive one and then applying linear filtering has been generalized and named homomorphic filtering [3]. In this case both multiplied terms (reflectance and transmission) are real and nonnegative so that the simple logarithm is an effective transform.

In order to follow the procedure outlined above, the sun illumination L must be estimated from the cloudy picture. Since a, $r(x,y)$, and $t(x,y)$ are all between 0 and 1, the maximum value of $s(x,y)$ cannot be greater than L (see Eq. 1). If the cloud transmission at any point is zero, the value of $s(x,y)$ at that point will be L. Therefore a reasonable value for L in a large set of consistent data is the brightest point present. To prevent computation problems deriving from the logarithm of zero, the value of L is set to one integer larger than the brightest point present. The original data is then processed by subtracting the intensity of each point from this estimate of L.

The logarithm is then taken of the inverted data. Now the signal and noise are additive (see Eq. 2). The filter function is

$$H(\mu,\nu) = \frac{S_{MP}(\mu,\nu)}{S_{PP}(\mu,\nu)} \tag{3}$$

where $S_{MP}(\mu,\nu)$ is the cross power spectrum between signal and signal-plus-noise and $S_{PP}(\mu,\nu)$ is the power spectrum of the signal-plus-noise. The two spatial frequency components are μ and ν. This is a non-causal filter which uses all the cloudy picture points to estimate each individual signal point [1]. In order to apply this filtering, an estimate must be made of $S_{MP}(\mu,\nu)$. This will be discussed in Section IV.

III. EXAMPLE OF HOMOMORPHIC FILTERING

An example of the ability of this filtering technique is now given to show its potential. A noisy picture is simulated using an original image $r(x,y)$ and a noise pattern $t(x,y)$ so that the output image $s(x,y)$ is formed by Eq. 1. Two-dimensional linear filtering is performed on $\log[L - s(x,y)]$ using known statistics of $r(x,y)$ and $t(x,y)$. The signal estimate is then obtained by exponentiating the filter output and inverting the grey levels. The complete process is shown in Fig. 2.

Results of several simulations using 64×64 pictures are shown. For comparison purposes, the mean and standard deviation of the noisy and filtered pictures are normalized so that $\mu = 128$ and $\sigma = 48$ on a display scale of 256. Figure 3(a) is a noisy signal with $L = 15$, $r(x,y) = 2/3$ in background and $4/5$ in foreground, and $t(x,y)$ is white noise, uniformly distributed between 0.02 and 1.0. The filtered result is shown in Fig. 3(b).

Figure 4 shows another noisy picture and the filtered results with the signal level decreased to 23/30 in the background and 4/5 in the foreground.

In Fig. 5(a) the same noise as used in Fig. 3(a) was low pass filtered before it was used. The signal edges (high frequencies) are retained in the filtered output because the noise has no components at these frequencies.

These simulation results give an idea of the maximum possible improvement because the model being used is exact and the noise statistics are known.

IV. ESTIMATION OF NOISE STATISTICS IN CLOUDY PICTURES

It is possible to assume the spatial and spectral properties of clouds can be modeled by a universal cloud model and use the statistics based on this model in the filter. However, a more accurate method is to estimate the cloud frequency content directly from the cloudy picture. This allows for wide variation in the type of clouds which can be removed. The basic assumptions made are that the clouds contain only low spatial frequencies and are located in the picture domain only where the intensity is above the normal received ground reflection.

Based on histogram data of cloudy and clear imagery, a threshold level for each spectral channel was selected which represented typical ground level reflections. Only points above this threshold level were considered as potential cloud points. These threshold settings were consistent across spectral channels so that approximately the same percentage of points in each channel were above threshold in a cloudy image.

The first approximation for cloud transmission in a particular image is

$$\hat{t}(x,y) = \frac{\hat{L} - s(x,y)}{\hat{L} - \hat{G}} \tag{4}$$

where $\hat{L}$ is the sun illumination estimate described earlier and $\hat{G}$ is the typical ground reflection [$a L r(x,y)$] estimate. The transmission estimate is bounded by $\frac{1}{\hat{L} - \hat{G}}$ and 1. Values falling outside this range are set to the

respective limits above. A transmission value of 1 is the theoretical maximum and although 0 is the theoretical minimum, computational problems occur with the logarithm operation if this is allowed.

The two-dimensional Fast Fourier Transform [4,5] of the logarithm of $\hat{t}(x,y)$ is then used to estimate the spatial frequency content of the clouds. The square of the magnitude of the transformed points represents the power spectral density of the noise components. However the high frequency parts of this power spectrum are removed using an ideal circularly-symmetric low pass filter of radius 9 cycles per picture width. This filter is used because our low frequency assumption about clouds implies that high frequency components, even though associated with high intensity picture points are more likely due to ground reflectance (concrete roads, etc.) than due to the noise (clouds). The resulting filtered version of the power spectrum is used as the actual noise power spectral estimate, $S_{NN}(\mu,\nu)$.

There are other possible methods of estimating the noise statistics which presently seem less promising. One method is to quantify the presence of clouds in the multispectral image using the LARS classifier [6]. This classifier processes multispectral data one point at a time classifying unknown data using training statistics developed from pre-classified data. Training classes can be chosen to include different percentage cloud cover.

Another method would measure statistics of clouds over water or some other essentially constant ground reflectance. In this case Eq. 2 reduces to

$$\log[L - s(x,y)] = \log[t(x,y)] + K \qquad (5)$$

where K is a constant, and the power spectrum obtained is that of the noise except for the d.c. (0,0) frequency point.

This power spectrum should be circularly symmetric since clouds have no preferred orientation. It should consist mainly of low spatial frequency components since clouds are relatively large and smooth functions compared to ground reflectance. Some care must be taken to normalize the power spectrum so that $\hat{M}(\mu,\nu)$ remains positive.

V. TWO-DIMENSIONAL WIENER FILTERING

Once the noise statistics are estimated, the Wiener filter of Eq. 3 can be implemented (see Fig. 2). The noise and signal are additive and can be assumed to be independent. However both signal and noise have nonzero means and these must be accounted for in the filter. If we assume the signal (M) and noise (N) are uncorrelated two-dimensional stationary random processes, the cross-correlation between the signal (M) and signal-plus-noise (P) is

$$R_{MP}(\tau_x,\tau_y) = E\{M(x+\tau_x,y+\tau_y)[M(x,y)+N(x,y)]\} \tag{6}$$

$$= R_{MM}(\tau_x,\tau_y) + \eta_M\eta_N$$

where η_M and η_N are the means of the signal and noise, respectively. The corresponding cross power spectrum is found from the two-dimension Fourier transform:

$$S_{MP}(\mu,\nu) = S_{MM}(\mu,\nu) + \eta_M\eta_N\,\delta(\mu,\nu) \tag{8}$$

where $\delta(\mu,\nu)$ is the two-dimensional direct delta function [4]. Similarly the spectral density

$$S_{PP}(\mu,\nu) = S_{MM}(\mu,\nu) + S_{NN}(\mu,\nu) + 2\eta_M\eta_N\,\delta(\mu,\nu) \tag{9}$$

Combining Eqs. 3, 8, and 9 results in the Wiener filter

$$H(\mu,\nu) = \frac{S_{PP}(\mu,\nu) - S_{NN}(\mu,\nu) - \eta_M \eta_N \delta(\mu,\nu)}{S_{PP}(\mu,\nu)} \tag{10}$$

Using the model resulting in Eq. 2, the term η_M is always positive and η_N is always negative. Therefore there is a boost at DC only and attenuation at frequencies where there is significant noise power. $S_{NN}(\mu,\nu)$ is estimated as described in Section IV. $S_{PP}(\mu,\nu)$ is estimated by the magnitude squared at the 2D FFT of $\log[L - s(x,y)]$ where L is estimated as described in Section II and $s(x,y)$ is the original cloudy picture.

It should be mentioned that the method of deriving the noise statistics discussed in Section IV slightly underestimates the noise power (due to the warping by Eq. 4). However, at some frequency points the noise estimate may be so high that Eq. 9 is negative, in which case, the filter function at that point is set to zero.

The filtered image

$$\hat{M}(\mu,\nu) = H(\mu,\nu)P(\mu,\nu) \tag{11}$$

is then retransformed as shown in Fig. 2 to obtain the ground reflectance estimate $aL\,r(x,y)$. Results using this technique are shown in Fig. 6. The upper left picture is a 256 x 256 cloudy LANDSAT image (channel 1) of the Chicago area. The upper right picture is the filtered output using the processing described above. The lower two pictures show results using an ideal high pass filter on the original cloudy data and using an ideal high pass filter on $\log[L - s(x,y)]$, respectively. The last two are included for comparison with the more accurate filtering based on the model.

VI. THREE DIMENSIONAL FILTERING

The real potential in the cloud filtering process is in incorporating a third dimension, the spectral channels, forming a three dimensional

reflection $r(x,y,z)$ and cloud transmission $t(x,y,z)$. The linear filter thus employed is three dimensional, $H(\mu,\nu,\rho)$ using three frequencies (two spatial and one spectral). Although there are only four points in the spectral dimension for LANDSAT data, the method has good promise, because most clouds follow the same response in the spectral dimension: cloud transmission increases with wavelength in a predictable fashion. This implies that the cloud transmission $t(x,y)$ in each spectral channel differs only by a multiplicative factor so that the noise, $\log[t(x,y)]$, differs in each channel by only an additive constant and the noise power spectrum in each spectral channel $S_{NN}(\mu,\nu)$ is identical except at each zero spatial frequency point. This implies that the three dimensional power spectrum of the noise

$$|3D\ FFT\{\log[t(x,y,z)]\}|^2 = S_{NN}(\mu,\nu,\rho) \tag{12}$$

should be zero for every point where both $\rho \neq 0$ and $(\mu,\nu) \neq (0,0)$. This coupled with the low frequency assumption of cloud content implies that the Wiener filter operates only on points where $\rho = 0$ and $\mu^2 + \nu^2 <$ (low pass filter radius)^2 and on the three points $\mu,\nu,\rho = 001, 002, 003$. Thus the effect of the transformation is to compress all cloud effects into a very small region where they can then be filtered out.

The actual filter estimate is made by performing a three dimensional FFT on the noise estimate $\log[\hat{t}(x,y,z)]$ and setting the appropriate regions in the frequency domain to zero. The filtering is then performed using three dimensional versions of Eqs. 10 and 11.

These results can be seen in Figs. 7-9. Figure 7 is all 4 spectral channels of the original cloudy image. Figure 8 shows the results of the three dimensional filtering. Figure 9 is an expanded version of the channel original and filtered result. The results are encouraging in that more

additional image details not present in the original cloudy image become visible and the filtered output is not significantly degraded.

VII. CONCLUSIONS

Two and three dimensional filtering of multispectral data to remove light cloud cover is a distinct possibility. In computer simulated noisy situations, the filtering results are good. Additional LANDSAT data should be processed to arrive at conclusive results as to the utility of such techniques. The measurement of noise statistics might be improved by iteratively using the estimated ground reflectance in place of the constant $\hat{G}$ in Eq. 4. The model of cloud distortion of images needs to be refined based on the results of filtering using the simple model presented. It may be necessary to consider convolutional effects of cloud cover as well as multiplicative effects. The change in multispectral classification accuracy after filtering may be a suitable measure of the performance of such homomorphic filters.

Although this discussion has been limited to cloud effects in satellite imagery, the techniques might be generalized to include the general case of the removal of interferring regions between a sensor and the object of interest as in seismic exploration.

REFERENCES

1. Papoulis, A., Probability, Random Variables, and Stochastic Processes, McGraw-Hill: New York, 1965.

2. Van Trees, H. L., Detection, Estimation, and Modulation Theory, Vol. 1, John Wiley: New York, 1968.

3. Oppenheim, A. V., Schafer, R. W., and Stockham, Jr., T. G., "Nonlinear Filtering of Multiplied and Convolved Signals," Proc. of the IEEE, Vol. 36, No. 8, pp. 1264-1291, Aug. 1968.

4. Rosenfeld, A. and Kak, A. C., Digital Picture Processing, Academic Press: New York, 1976.

5. Gold, B. and Radar, C. M., Digital Processing of Signals, McGraw-Hill: New York, 1969.

6. Lindenlaub, J. C., "Guide to Multispectral Data Analysis Using LARSYS," LARS Information Note 062874, LARS, Purdue University, West Lafayette, Indiana, 1974.

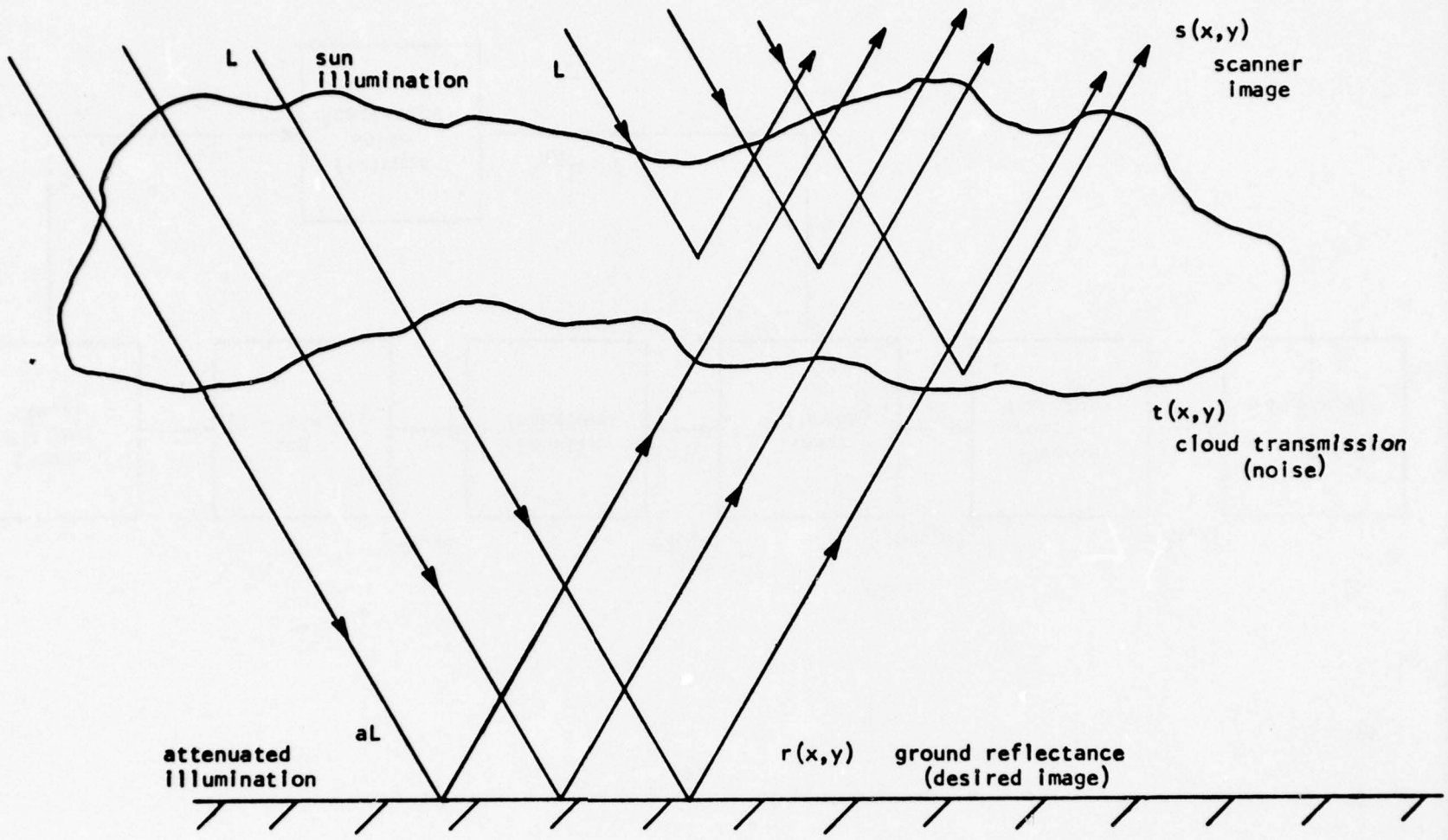


Figure 1 Satellite Scanner Image Formation.

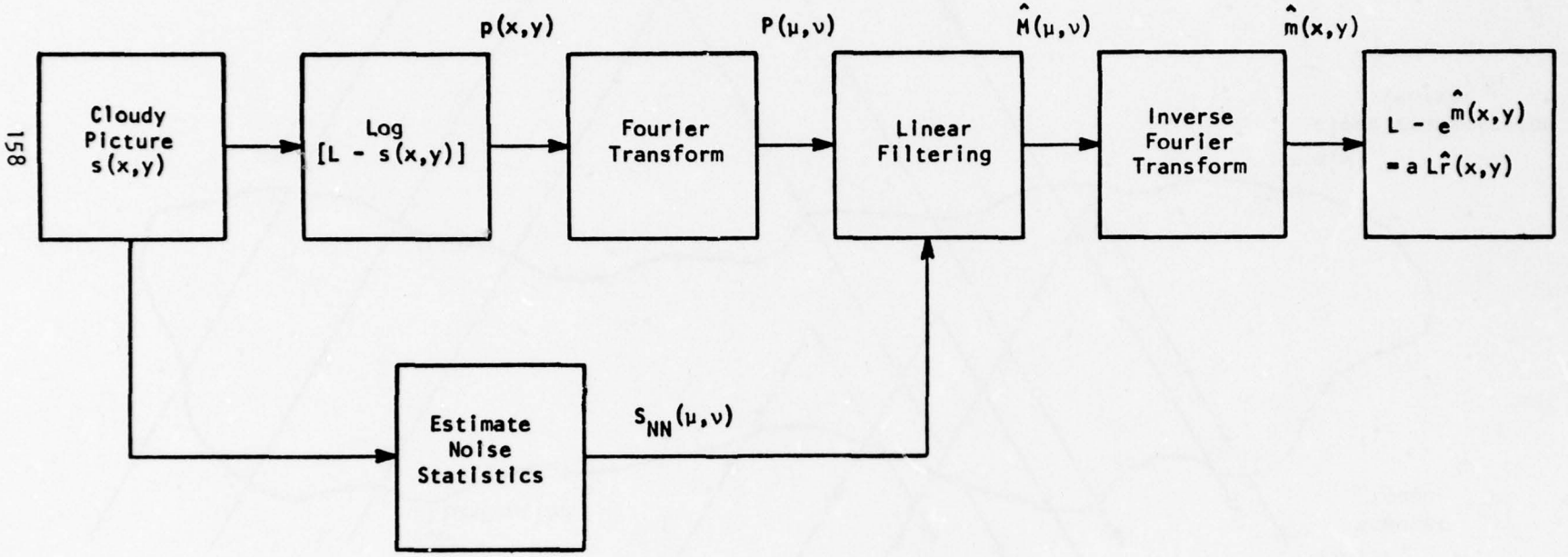


Figure 2 Homomorphic Filtering Process.

(a)

(b)

Figure 3 Computer Simulated Noisy Image (a) and Filtered Result (b). Signal and Noise Follow Model in Eq. 1; $L = 15$; $r(x,y) = 2/3$ in Background and 4/5 in Foreground; $t(x,y)$ is Uniformly Distributed Between 0.02 and 1.0. Filtering Process is as Shown in Figure 2.

(a)

(b)

Figure 4 Computer Simulated Noisy Image (a) and Filtered Result (b). Same as Figure 3 Except $r(x,y)$ Ranges from 0.767 and 0.800.

(a)

(b)

Figure 5 Computer Simulated Noisy Image (a) and Filtered Result (b). Same as Figure 3 Except $t(x,y)$ is Low Pass Filtered so that the Maximum Noise Frequency is 8 Cycles/Picture Width.

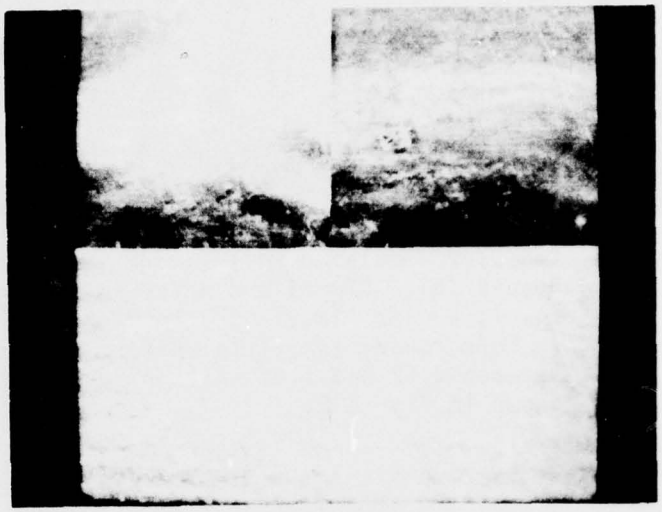


Figure 6 Two Dimensional Filtering Results. Upper Left - Original Landsat Image, 256 x 256, Channel 1 (Green); Upper Right - Output of Filter Described in Text; Lower Left - Output of Ideal Low Pass 2D Filter on Original; Lower Right - Output of Ideal Low Pass 2D Filter on log[L - s(x,y)].

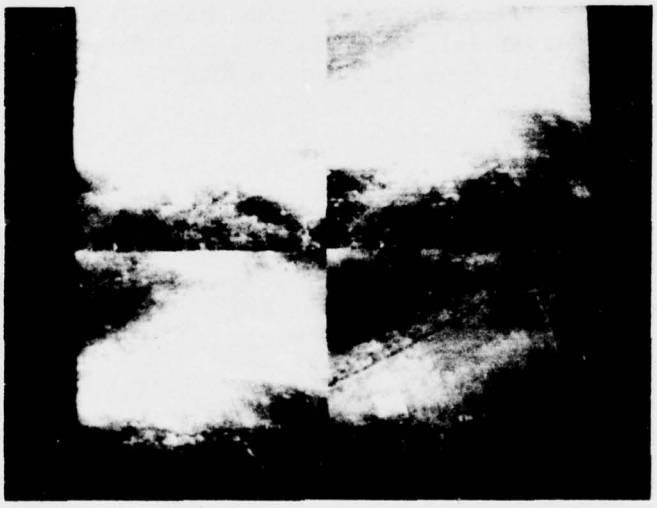


Figure 7 Original Landsat Data. Upper Left - Channel 1 (Green); Upper Right - Channel 2 (Red); Lower Left - Channel 3 (Infrared); Lower Right - Channel 4 (Infrared).

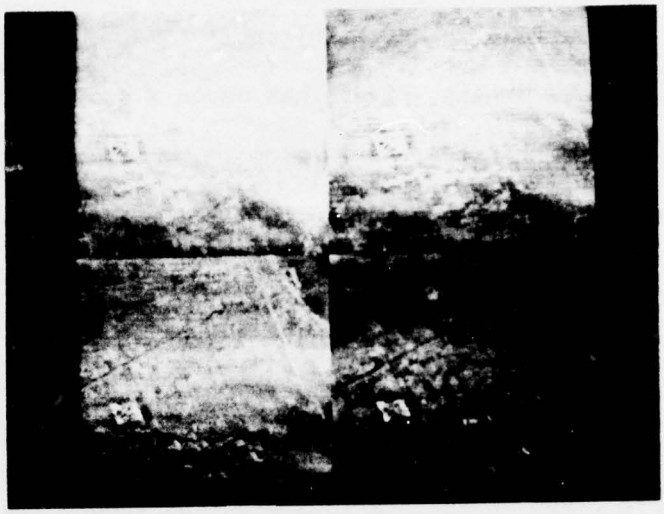


Figure 8 Three Dimensional Filtering Results. Channels are Arranged as in Fig. 7.

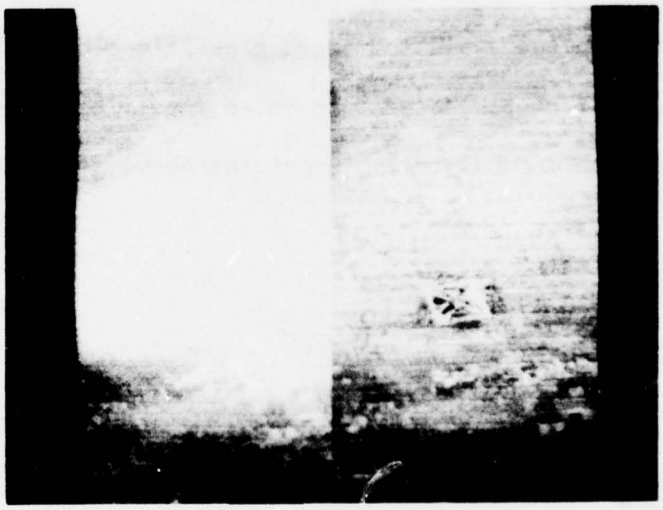


Figure 9 Enlarged Section of Channel 1 from Figs. 7 and 8.

FLIR IMAGERY TACTICAL TARGET DETECTION AND CLASSIFICATION

O.R. Mitchell

Over the last few months, Purdue has begun a project in cooperation with the Honeywell System and Research Division in Minneapolis, Minnesota. The goal is to develop a second-generation real-time target cueing system for FLIR (forward looking infrared) imagery.

Purdue's contribution is feature evaluation, feature subset selection, and an intelligent combination of simple measurements to allow fast, reliable target detection and classification. An initial discussion of this problem and typical data were given in the previous progress report (pp. 129-133).

Work now in progress is:

(1) Collect a reasonable data set of FLIR imagery to allow comparison of methodologies and classification accuracy.

(2) Compare image decomposition methods-both statistical or syntactic.

(3) Develop and evaluate target classification methods which use structure and texture in an intelligent manner.

(4) Develop methods to implement the above algorithms in real time.

COMPUTER FACILITY DEVELOPMENTS

A. P. Reeves

During the last year the changeover to the UNIX operating system has been completed. Now UNIX is the only operating system being run on the PDP-11/45 computer. UNIX was originally installed for two main reasons: Firstly, a network version of UNIX for interfacing with the ARPANET was being developed at University of Illinois. This would give us a better software interface than the ELF system which we were then using. Secondly, even though UNIX has been available for several years now, it is still probably the most advanced operating system in common use on a minicomputer. It has a well-planned file structure and contains features which are usually only found in much larger computer systems. Converting to it was a major decision because it meant discarding all the considerable amount of software which had been developed up to that time, for communicating with the ARPANET and for driving the Comtal TV display.

The research work has been divided between three computers: our own PDP-11, the IBM-360 at the Laboratory for the Application of Remote Sensing (LARS) and the two CDC-6500 computers of the Purdue University Computing Center. The intent originally was to establish the PDP-11 as a very intelligent terminal to the ARPANET, then the work being done on the other local computers would be transferred to remote ARPANET computers. The initial work on the UNIX system was to write software to control the Comtal TV display and to read data from magnetic tapes produced by the other two computers.

We have had considerable problems with the hardware on the PDP-11. A capacitor disintegrating in one of the magnetic tape drives also destroyed part of the processor; this resulted in the system being down for about two months. A more serious problem is the intermittent faults which have occurred

removed. This will enable pictures and large programs to be much more conveniently stored off-line on disks rather than magnetic tape. Finally, the overall capacity of the system is increased.

Several minor hardware projects are also progressing. These include: installation of a paper tape reader, a faster interface for the Versatek plotter, a fast 1 bit per pixel TV display, and a program controlled switch box for better interactive communication with the Comtal display and some programs.

Several software projects have also been started. An inovative scheme for maintaining compatable sets of picture data, called PDS, is being implemented. The initial work on PDS will be to design a file header and a set of picture formats to be used by everyone in the laboratory. A set of user callable system subroutines will be written to maintain and access these files. Presently a provisional data format has been designed and the system subroutines will be made available to users within the next quarter. The next phase of development will be to organize picture files into data structures utilizing the flexible, tree file structure which is the basis of the UNIX operating system.

Some utility programs will be completed during the next quarter, these include the following: Firstly, a documentation program, which will enable users to very rapidly access, on-line, any of the documentation which exists for the system and the user developed software. Also a program has been written to assist users in documenting their own software with a standard format. Secondly, a graphics package, which will enable users to draw lines, shapes and text onto the devices with graphics capabilities: these include the Comtal TV display, the Versatek plotter, the Tektronix 4010 terminal, and the line printer.

on the RP03 disk. This disk is the heart of the computer system and nothing will run without it. In the summertime these intermittent faults would cause one or two major system crashes a week. Several times these faults were due to component failure but for the others we still do not know the cause. The software for dumping and restoring the disk files has been considerably extended so that recovery with user files intact, from major system crashes can be made much more quickly.

At the end of October, the PDP-11 was disconnected from the ARPANET. The role of the computer has therefore changed, to provide the local computing facilities for all the research projects. Fortunately, the UNIX operating system provides an excellent foundation for an image processing facility, however, considerable software development is required. For example, the main high level language for UNIX is 'C' which is much better organized than FORTRAN. However, much of the work on other computers has been done in FORTRAN. The standard UNIX FORTRAN is rather rudimentary and conversion problems occur when transferring programs from the larger computer installations.

During the last quarter of this year considerable new development of the computer system has been initiated. The construction of a TV camera interface has been started. This will enable us to digitize our own pictures in many cases, rather than rely entirely on other computer installations. We intend to purchase two RK11 disk drives for the following reasons: Firstly, they will enable the system to run when there is trouble with the RP03 disk. Currently the RP03 disk is responsible for most of the problems that occur with the computer system. Secondly, the efficiency of the system will be improved, especially when transferring data from one disk file to another is required. Thirdly, the RK05 cartridge disks may be quickly loaded and

During the next quarter we intend to transfer some standard image classification programs developed at LARS to the PDP-11. Thiw will be started when the system subroutines for accessing PDS picture files are available. These programs will be especially useful to researchers who started their work on the LARS computer.

Currently we have seven time-sharing terminals which may be accessed by users at any hour of the day. Also the Electrical Engineering Department has recently had a PDP-11/70 installed; this computer is running the same version of UNIX that is run on our PDP-11/45. The ARPA researchers have limited access to this computer and, in emergency situations involving difficulty with our own system, they would be able to use it for program development.

FACILITIES

QTY	Manufacturer	Description
3	Beehive Elect.	"Super-Bee" Terminals
2	Tex. Inst.	"Silent 700" Terminals
1	Digi-Data	Industry standard magnetic tape system; 2, 9-track and 1, 7-track drives; one each NRZI and phase-encoded formatters/controllers
1	DEC	Dual-drive DEC tape unit
1	DEC	RP03 disk drive (40 million characters)
1	Fabritek	96K-word auxiliary memory system (64K bought by ARPA, 32K by NASA)
1	Versatek	Electrostatic matrix printer
1	Comtal	Color picture display
1	Data Printer	132 column, 600 L.P.M. line printer
1	True-Data	Punched card reader
1	Tektronix	Model 4010, graphics display
1	DEC	PDP-11/45 computer system; system includes: 32K memory FPP-11 floating point processor (NSF money) H960 extension mounting cabinet 3 - small peripheral mountings blocks (DD-11) 1 UNIBUS repeater/expander DH11, 16-line terminal multiplexor KW11-p programmed clock

PUBLICATIONS

BOOKS

FU, K.S., "ERTS Data Analysis," chapter in Applications of Syntactic Pattern Recognition, ed., Springer-Verlag, 1976. (with J. Brayer, P. H. Swain)

HUANG, T., Image Processing and Digital Filtering (editor), Springer-Verlag, 1975.

SWAIN, P.H., "ERTS Data Analysis," chapter in Applications of Syntactic Pattern Recognition, K.S. Fu, ed., accepted for publication by Springer-Verlag, 1976.

WINTZ, P.A., "Picture Coding and Feature Extraction," chapter in Digital Image Processing, 1976.

JOURNAL PUBLICATIONS

FU, K.S., "Pattern Recognition in Remote Sensing of the Earth's Resources," IEEE Trans. on Geoscience Electronics, January 1976.

FU, K.S., "Error Estimation in Pattern Recognition via L^d-Distance Between Posteriori Density Functions," IEEE Trans. on Information Theory, Vol. IT-22, pp. 43-52, January 1976. (with T. Lissack)

FU, K.S., "A Tree System Approach for Fingerprint Pattern Recognition," IEEE Transactions on Computers, Vol. C-25, pp. 262-274, March 1976. (with B. Moayer)

FU, K.S., "An Application of Stochastic Languages to Fingerprint Pattern Recognition," Pattern Recognition, Vol. 8, pp. 175-181, 1976. (with B. Moayer)

FU, K.S., "A Minicomputer Facility for Picture Processing and Pattern Recognition Research" COMPUTER, Vol. 9, pp. 70-77, May 1976. (with E. Persoon)

FU, K.S., "Parametric Feature Extraction Through Error Minimization, Applied to Medical Diagnosis," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-6, September 1976. (with T. Lissack)

FUKUNAGA, K., "A Graph-Theoretic Approach to Nonparametric Cluster Analysis," IEEE Trans. on Computers, Vol. C-25, pp. 936-944, Sept. 1976. (with W.L.G. Koontz, P.M. Narendra)

HUANG, T., "The Importance of Phase in Image Processing Filters," IEEE Trans. on Acoustics, Speech, and Signal Processing, December 1975. (with J. Burnett and A. Deczky)

HUANG, T., "Facsimile Coding by Skipping White," IEEE Trans. on Comm., December 1975. (with A. Hussian)

MITCHELL, O.R., "Effect of Spatial Frequency on the Visibility of Unstructured Patterns," J. Opt. Soc. Am., Vol. 66, No. 4, April 1976.

MITCHELL, O.R., "Digital Communications Equipment for Instructional Purposes," IEEE Trans. on Education, May 1976. (with W. L. Thomas)

MITCHELL, O.R., "Hemispheric Specialization and Cognitire Development," accepted for publication in the Journal for Research in Mathematics Education. (with Grayson H. Wheatley and R. L. Frankland)

MITCHELL, O.R., "A Max-Min Measure for Image Texture Analysis," accepted for publication in IEEE Transactions on Computers. (with C.R. Myers)

MITCHELL, O.R., "Filtering to Remove Cloud Cover in Satellite Imagery", to be published in IEEE Transactions on Geoscience Electronics. (with E.J. Delp and P.L. Chen)

SWAIN, P.H., "Determining Density of Maize Canopy from Digitized Photographic Data," Agronomy Journal, Vol. 68, pp. 55-59, January-February 1976. (with E.R. Stoner and M.F. Baumgardner)

SWAIN, P.H., "Selective Radiant Temperature Mapping Using a Layered Classifier," IEEE Trans. Geoscience Electronics, Vol. GE-14, pp. 101-106, April 1976. (with L.A. Bartolucci and C.L. Wu)

CONFERENCES

FU, K.S., "Processing of Chest X-Ray Images by Computer," IFIP Working Conference on Decision-Making and Medical Care, May 24-29, 1976, Dijon, France.

FU, K.S., "A Syntactic Approach to the Representation of Image Structures," IFIP Working Conference on Environmental Modelling, April 26-28, 1976, Tokyo, Japan.

FU, K.S., "High Dimensional Languages and Grammatical Inference," IEEE Joint Workshop on Pattern Recognition and Artificial Intelligence, June 1-3, 1976, Hyannis, MA.

FU, K.S., "Some Applications of Stochastic Languages," Symposium on Application of Statistics, June 14-18, 1976, Dayton, Ohio.

FU, K.S., "Tree System Approach for LANDSAT Data Interpretation," Proc. Symp. on Machine Processing of Remotely Sensed Data, June 29-July 1, 1976.

FU, K.S., "An Approach to the Design of a Linear Binary Tree Classifier," Proc. Symp. on Machine Processing of Remotely Sensed Data, June 29-July 1, 1976.

FU, K.S., "The Linguistic Approach to Pattern Recognition," Advanced Seminar on Classification and Clustering, The Mathematics Research Center, University of Wisconsin, Madison, Wisconsin, May 3-5, 1976.

FU, K.S., "Image Processing Algorithms Applied to Rib Boundary Extraction in Chest Radiographs," Engineering Foundation Conference on Algorithms for Image Processing, August 8-13, 1976, Rindge, N.H.. (co-author)

FUKUNAGA, K., "Diagnostic Signature Analysis of Small Rotating Machines," 1976 Milwaukee Symposium on Automatic Computation and Control, Milwaukee, Wisconsin, April 1976. (with T. Usami and T. Koizumi)

FUKUNAGA, K., "A Graph-Theoretic Approach to Nonparametric Cluster Analysis," 1976 IEEE International Symposium on Information Theory, Ronneby, Sweden, June 1976. (with W.L.G. Koontz and P.M. Narendra)

HUANG, T., "Image Processing Research at Purdue University," USSR-US Joint Workshop on Inf. Theo., December 14-19, 1975, Moscow, U.S.S.R.

HUANG, T., "Facsimile Coding by Skipping White," presented at the 1976 Picture Coding Symposium, Asilomar, January 28-30, 1976.

HUANG, T., "Restoration of Images Degraded by Spatially-Varying Systems," presented at the OSA Technical Meeting on Image Processing, Asilomar, February 24-25, 1976.

HUANG, T., "Some Experience in Stability Tests for Two-Dimensional Recursive Digital Filters," Arden House Workshop on Digital Signal Processing, Feb. 23-26, 1976, Hamison, NY. (with B.T. O'Connor)

HUANG, T., and FU, K.S., "Research on Image Understanding at Purdue," ARPA Workshop on Image Understanding, April 12-13, 1976, Univ. of Southern California, Los Angeles.

HUANG, T., organizer and lecturer, short courses on "Digital Techniques in Spectral Analysis, Estimation and Filtering," March 8-12, 1976, and "Image Processing and Pattern Recognition," April 12-16, 1976, Purdue University.

HUANG, T., "Image Processing Research at Purdue," presented at Los Alamos Scientific Laboratory, Los Alamos, NM, May 19, 1976.

HUANG, T., "Digital Straight Edges," presented at the 6th Annual Symp. on Automatic Imagery Pattern Recognition, Univ. of Maryland, Silver Spring, MD, June 1-2, 1976. (with G. Tang)

HUANG, T., "Two-Dimensional Fourier Transforms," "Image Restoration," and "Film Models," presented at NATO Advanced Institute on Digital Image Processing and Analysis, Bonas, France, June 14-25, 1976.

HUANG, T., and MITCHELL, O.R., "Subjective Effect of Two-Dimensional Noise," SPSE, Symp. on Image Evaluation, July 19-23, 1976, Toronto, Canada.

MITCHELL, O.R., "An EEG Investigation of the Differences in the Hemispheric Specialization of Formal and Concrete Operational Persons," presented at the 1976 Annual Meeting of the American Educational Research Association, San Francisco, CA, April 1976. (with R.A. Dilling and G.H. Wheatley)

MITCHELL, O.R., "Texture Edge Detection and Classification Using Max-Min Descriptors," Sixth Annual Symposium on Automatic Imagery Pattern Recognition, Univ. of Maryland, College Park, MD, June 1-2, 1976.

MITCHELL, O.R., "Filtering to Remove Cloud Cover in Satellite Imagery," LARS Symposium, Machine Processing of Remotely Sensed Data, June 29-July 1, 1976, West Lafayette, IN. (with P.L. Chen)

MITCHELL, O.R., and HUANG, T., "Subjective Effect of Two-Dimensional Noise," SPSE Symp. on Image Evaluation, July 19-23, 1976, Toronto, Canada.

MITCHELL, O.R., "Image Noise Visibility and Fidelity Criteria," Electro-Optical Systems Design Conference/International Laser Exposition, New York, N.Y., September 14-16, 1976. (with E.J. Delp)

SWAIN, P.H., "Application of a Class of Sequential Classifiers to Multitemporal Remote Sensing Data," Proc. Third Symposium on Machine Processing of Remotely Sensed Data, June 1976. (with H. Hauska)

SWAIN, P.H., "Some Time for Texture in the Spectrum of Spatial Features," presented at the Engineering Foundation Conference on Algorithms for Image Processing, Franklin Pierce College, Rindge, N.H. August 1976. (with D.A. Landgrebe)

SWAIN, P.H., "Applications of Image Processing in Remote Sensing: Can We Get It All Together?" session summary and discussion, Second Workshop on Advanced Automation (NSF), University of Maryland, College Park, MD, October 1976.

WINTZ, P.A., "Digital Telephony and Signal Processing Theory and Practice," Instanbul, Turkey, December 1975. (with J. Sergo)

WINTZ, P.A., "Images and Models for Image Noise," presented at NATO Advanced Institute on Digital Image Processing and Analysis, Bonas, France, June 14-25, 1976.

WINTZ, P.A., "Image Coding with Emphasis on Techniques for Producing Decorrelated Image Data," presented at NATO Advanced Institute on Digital Image Processing and Analysis, Bonas, France, June 14-25, 1976.

STAFF

CO-PRINCIPAL INVESTIGATORS

T. S. Huang
K. S. Fu

PROFESSORIAL

K. Fukunaga
O. Mitchell
A. Reeves
P. Swain
P. Wintz

GRADUATE RESEARCHERS

S. Berger
J. Burnett
S. Carlton
Wm. Chan
P. H. Chen
P. L. Chen
X. Dang
E. Delp
R. Florek
J. Keng
E. Kit
R. L. Li
Y. K. Lin
P. Narendra
B. O'Connor
D. Panda
W. Pfaff
A. Salahi
R. Short
G. Tang
T. Wallace
M. Yoo
K. You
T. S. Yu

RESEARCH STAFF

J. Besemer
W. Robey

UNDERGRADUATE RESEARCHERS

C. Buckwacter
M. DeMoney
T. Hawker
R. Johnson
J. Meese
J. Schwab
B. Zurney

ELECTRONIC TECHNICIANS

D. Azpell
P. Crane
J. Rogers
F. Woodworth

SECRETARIES

M. Barbour
M. Claire

METRIC SYSTEM

BASE UNITS:

Quantity	Unit	SI Symbol	Formula
length	metre	m	...
mass	kilogram	kg	...
time	second	s	...
electric current	ampere	A	...
thermodynamic temperature	kelvin	K	...
amount of substance	mole	mol	...
luminous intensity	candela	cd	...

SUPPLEMENTARY UNITS:

plane angle	radian	rad	...
solid angle	steradian	sr	...

DERIVED UNITS:

Acceleration	metre per second squared	...	m/s
activity (of a radioactive source)	disintegration per second	...	(disintegration)/s
angular acceleration	radian per second squared	...	rad/s
angular velocity	radian per second	...	rad/s
area	square metre	...	m
density	kilogram per cubic metre	...	kg/m
electric capacitance	farad	F	A·s/V
electrical conductance	siemens	S	A/V
electric field strength	volt per metre	...	V/m
electric inductance	henry	H	V·s/A
electric potential difference	volt	V	W/A
electric resistance	ohm		V/A
electromotive force	volt	V	W/A
energy	joule	J	N·m
entropy	joule per kelvin	...	J/K
force	newton	N	kg·m/s
frequency	hertz	Hz	(cycle)/s
illuminance	lux	lx	lm/m
luminance	candela per square metre	...	cd/m
luminous flux	lumen	lm	cd·sr
magnetic field strength	ampere per metre	...	A/m
magnetic flux	weber	Wb	V·s
magnetic flux density	tesla	T	Wb/m
magnetomotive force	ampere	A	...
power	watt	W	J/s
pressure	pascal	Pa	N/m
quantity of electricity	coulomb	C	A·s
quantity of heat	joule	J	N·m
radiant intensity	watt per steradian	...	W/sr
specific heat	joule per kilogram-kelvin	...	J/kg·K
stress	pascal	Pa	N/m
thermal conductivity	watt per metre-kelvin	...	W/m·K
velocity	metre per second	...	m/s
viscosity, dynamic	pascal-second	...	Pa·s
viscosity, kinematic	square metre per second	...	m/s
voltage	volt	V	W/A
volume	cubic metre	...	m
wavenumber	reciprocal metre	...	(wave)/m
work	joule	J	N·m

SI PREFIXES:

Multiplication Factors	Prefix	SI Symbol
1 000 000 000 000 = 10^{12}	tera	T
1 000 000 000 = 10^{9}	giga	G
1 000 000 = 10^{6}	mega	M
1 000 = 10^{3}	kilo	k
100 = 10^{2}	hecto*	h
10 = 10^{1}	deka*	da
0.1 = 10^{-1}	deci*	d
0.01 = 10^{-2}	centi*	c
0.001 = 10^{-3}	milli	m
0.000 001 = 10^{-6}	micro	μ
0.000 000 001 = 10^{-9}	nano	n
0.000 000 000 001 = 10^{-12}	pico	p
0.000 000 000 000 001 = 10^{-15}	femto	f
0.000 000 000 000 000 001 = 10^{-18}	atto	a

* To be avoided where possible.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.